

Secondary Data Support and Non-Homogeneities  
in  
Space-Time Adaptive Processing

THESIS  
Todd Benjamin Hale  
Captain, USAF

AFIT/GE/ENG/97D-06

Approved for public release; distribution unlimited.

Research sponsored in part by the Rome Laboratory, Air Force Materiel Command, USAF, through the use of the MCARM data under a cooperative agreement number. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Rome Laboratory or the U.S. Government.

19980127 025

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

Secondary Data Support and Non-Homogeneities  
in  
Space-Time Adaptive Processing

THESIS

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University  
In Partial Fulfillment of the  
Requirements for the Degree of  
Master of Science in Electrical Engineering

Todd Benjamin Hale, B.S.E.E.

Captain, USAF

December, 1997

Approved for public release; distribution unlimited.

Research sponsored in part by the Rome Laboratory, Air Force Materiel Command, USAF, through the use of the MCARM data under a cooperative agreement number. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Rome Laboratory or the U.S. Government.

Secondary Data Support and Non-Homogeneities

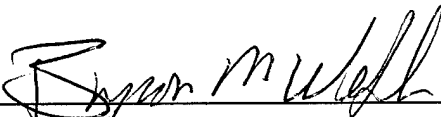
in

Space-Time Adaptive Processing

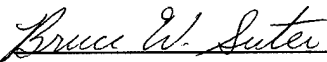
Todd Benjamin Hale, B.S.E.E.

Captain, USAF

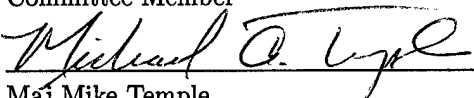
Approved:

  
\_\_\_\_\_  
Dr. Byron Welsh  
Thesis Advisor

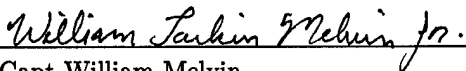
29 Oct 97  
Date

  
\_\_\_\_\_  
Dr. Bruce Suter  
Committee Member

29 October 1997  
Date

  
\_\_\_\_\_  
Maj Mike Temple  
Committee Member

29 Oct 97  
Date

  
\_\_\_\_\_  
Capt William Melvin  
Committee Member

29 Oct 97  
Date

### *Acknowledgements*

I would like to express my sincere thanks to Dr. Byron Welsh for his guidance throughout this project. He rose above the professional challenges in his life to meet the needs of his students.

Furthermore, I extend my appreciation to my committee. Dr. Mike Temple's knowledge of radar provoked many interesting discussions while reviewing the thesis. His comments kept me from falling into the many pitfalls. Dr. Bruce Suter's expertise in signal processing proved very valuable as well. I also thank Dr. Bill Melvin for the support I recieved from Rome Laboratory. Without his input, this project would not have been possible.

## *Table of Contents*

	Page
List of Figures . . . . .	vii
List of Tables . . . . .	xi
List of Symbols . . . . .	xii
List of Abbreviations . . . . .	xviii
Abstract . . . . .	xix
 I. Introduction . . . . .	 1-1
1.1 Background . . . . .	1-1
1.2 Objective . . . . .	1-2
1.3 Material and Equipment . . . . .	1-3
1.4 Support . . . . .	1-3
1.5 Organization . . . . .	1-3
 II. General Beamforming Concepts . . . . .	 2-1
2.1 Introduction . . . . .	2-1
2.2 Foundations . . . . .	2-1
2.2.1 Wavenumber-Frequency Space . . . . .	2-3
2.2.2 Space-Time Signal Decomposition . . . . .	2-4
2.2.3 Wavenumber-Frequency Space Translation . . . . .	2-5
2.3 Weighted Delay and Sum Beamformer . . . . .	2-7
2.3.1 Beam Steering Through Time Delays . . . . .	2-9
2.3.2 Antenna Array Pattern . . . . .	2-12
2.3.3 Linear Antenna Array Example . . . . .	2-15
2.3.4 Beam Shaping Through Windowing . . . . .	2-17
2.4 Summary . . . . .	2-21

	Page
III. Space-Time Adaptive Processing Primer . . . . .	3-1
3.1 Introduction . . . . .	3-1
3.2 The Physical and Electromagnetic Environment . . . . .	3-2
3.2.1 Physical Geometry . . . . .	3-2
3.2.2 Signal and Radar Description . . . . .	3-4
3.3 Fundamentals of STAP . . . . .	3-10
3.3.1 Thermal Noise Model . . . . .	3-13
3.3.2 Jamming Model . . . . .	3-14
3.3.3 Clutter Model . . . . .	3-21
3.4 Direct Form Fully Adaptive STAP . . . . .	3-38
3.5 Partially Adaptive Factored Time-Space STAP . . . . .	3-50
3.6 Summary . . . . .	3-55
IV. Reduced Dimensionality Through the Cross Spectral Metric Algorithm . . . .	4-1
4.1 Introduction . . . . .	4-1
4.2 Fundamentals . . . . .	4-1
4.3 Translating Direct Form STAP into a Generalized Sidelobe Canceler Form . . . . .	4-4
4.4 Summary . . . . .	4-29
V. Covariance Matrix Estimation and Non-homogeneity Detection . . . . .	5-1
5.1 Introduction . . . . .	5-1
5.2 Covariance Matrix Estimation . . . . .	5-2
5.3 Non-Homogeneity Detection . . . . .	5-4
5.4 MCARM Example . . . . .	5-7
5.5 Summary . . . . .	5-18

	Page
VI. Analysis of the CSM Algorithm . . . . .	6-1
6.1 Introduction . . . . .	6-1
6.2 SINR Comparisons for Known Covariance . . . . .	6-2
6.2.1 Fully Adaptive . . . . .	6-2
6.2.2 Partially Adaptive . . . . .	6-6
6.3 Secondary Data Support for Covariance Estimation . . . . .	6-9
6.3.1 Hypothesis Behind Poor Performance . . . . .	6-22
6.4 Non-Homogeneities and the CSM Algorithm . . . . .	6-27
6.5 Summary . . . . .	6-32
VII. Conclusions and Recommendations . . . . .	7-1
7.1 Conclusions . . . . .	7-1
7.2 Recommendations for Future Research . . . . .	7-3
Appendix A. Kronecker Product . . . . .	A-1
Appendix B. CSM Blocking Matrix Determination . . . . .	B-1
Appendix C. Performance Measures . . . . .	C-1
C.1 RMB Test . . . . .	C-1
C.2 GLRT Test . . . . .	C-3
C.3 AMF CFAR Test . . . . .	C-3
Bibliography . . . . .	BIB-1
Vita . . . . .	VITA-1
Index . . . . .	INDEX-1



## *List of Figures*

Figure		Page
2.1.	Wavenumber-Frequency space representation of space-time signals with the same frequency. . . . .	2-5
2.2.	Wavenumber-Frequency space representation of space-time signals with the same velocity of propagation. . . . .	2-5
2.3.	Wavenumber-Frequency space representation of space-time signals with the same direction of propagation. . . . .	2-5
2.4.	The intersection of the constant velocity of propagation cone and the directional plane in wavenumber-frequency space. . . . .	2-7
2.5.	A block diagram of the signal environment. . . . .	2-8
2.6.	Block diagram of the weighted delay and sum beamformer. The output of the beamformer as a function of time is designated $bf(t)$ . . . . .	2-9
2.7.	Calculation of the delay required in each element to steer the beam. . . . .	2-11
2.8.	Antenna pattern magnitude for the weighted delay and sum beamformer, 10 elements spaced 0.1m apart, and no beam steering. . . . .	2-17
2.9.	Antenna pattern magnitude for the weighted delay and sum beamformer, 10 elements spaced 0.1m apart, with beam steering. . . . .	2-18
2.10.	Pictorial representation of one dimensional rectangular and Hamming windows used for the receiver weights. . . . .	2-19
2.11.	Hamming window applied to the weights of a linear antenna array of 10 elements spaced 0.1m apart. . . . .	2-20
3.1.	(a) Uniformly spaced antenna element array geometry and (b) radar platform geometry. . . . .	3-3
3.2.	Time domain and frequency domain plots of the transmitted pulse illustrating ambiguous Doppler frequency. . . . .	3-7
3.3.	Antenna array element block diagram. . . . .	3-8
3.4.	The CPI datacube of dimension $M \times N \times L$ . . . . .	3-13
3.5.	The jammer bandwidth time domain correlation relationship. . . . .	3-16

Figure		Page
3.6.	Top and side views of the clutter ring. . . . .	3-22
3.7.	Clutter ridge plots illustrating the normalized Doppler aliasing for various values of $\beta$ . . . . .	3-30
3.8.	Correlated clutter sampling ( $\beta$ equal to an integer) gives low rank. . . . .	3-31
3.9.	Element voltage pattern for the clutter example. . . . .	3-33
3.10.	CNR around the clutter ring (main beam pointed at $0^\circ$ azimuth). . . . .	3-37
3.11.	Illustration of Brennan's rule for clutter matrix rank using various values of $\beta$ . . . . .	3-39
3.12.	An SINR comparison of fully adaptive STAP for the Blackman Harris, Hanning, and Hamming [11] tapered steering vectors. . . . .	3-43
3.13.	An SINR loss comparison of fully adaptive STAP for the Blackman Harris, Hanning, and Hamming [11] tapered steering vectors. . . . .	3-45
3.14.	Top level block diagram for a space-time adaptive processor. . . . .	3-45
3.15.	Fully adaptive space-time processing model. . . . .	3-46
3.16.	Fully adapted antenna pattern at 100Hz Doppler frequency. . . . .	3-47
3.17.	Fully adapted antenna pattern at $0^\circ$ azimuth. . . . .	3-48
3.18.	Three dimensional image of the fully adapted pattern $P_w$ with no tapering for a single target and two jammers. . . . .	3-49
3.19.	The Factored Time Space (FTS) reduced dimension algorithm block diagram. . . . .	3-50
3.20.	Comparison of $L_{\text{SINR}}$ for the FTS algorithm with four Doppler Filter bank windows and no spatial taper. . . . .	3-54
3.21.	Comparison of $L_{\text{SINR}}$ for the FTS algorithm with four Doppler Filter bank windows and the same spatial window as the Doppler window. . . . .	3-55
4.1.	The direct form adaptive processor block diagram showing the optimum constrained filter weights. . . . .	4-6
4.2.	A block diagram of the blocking matrix concept used in the formulation of the GSC model of STAP. . . . .	4-7
4.3.	The full dimension GSC block diagram. . . . .	4-9
4.4.	The full dimension GSC block diagram with eigenvector projection. . . . .	4-13
4.5.	The reduced dimension GSC block diagram using $O$ eigenvectors as a projection. . . . .	4-14

Figure		Page
4.6.	A comparison of the reduced dimensionality SINR for the principal components and cross spectral metric methods. . . . .	4-28
5.1.	Non-homogeneity detection example: Beamformed Blackman Harris Doppler filter output by bin. . . . .	5-9
5.2.	Non-homogeneity detection example: Beamformed Blackman Harris Doppler filter output by range cell. . . . .	5-11
5.3.	Non-homogeneity detection example: FTS algorithm output and AMF CFAR test. . . . .	5-12
5.4.	Non-homogeneity detection example: Secondary data vector ordering by homogeneity. . . . .	5-14
5.5.	Non-homogeneity detection example: Secondary data vector ordering by non-homogeneity. . . . .	5-16
5.6.	Non-homogeneity detection example: The FTS algorithm output using the GIP non-homogeneity detector. . . . .	5-17
6.1.	A plot of the fully adaptive SINR expression of the CSM algorithm versus normalized Doppler. . . . .	6-4
6.2.	Reduced dimension SINR using the cross spectral metric compared to full dimensionality. . . . .	6-8
6.3.	The experimental SINR curve versus secondary data set size for the FTS algorithm. . . . .	6-12
6.4.	The experimental $L_{\text{SINR}}$ curve versus secondary data set size for the full dimensioned CSM algorithm. . . . .	6-14
6.5.	The optimum SINR curve (known covariance) versus weight vector dimension $O$ . . . . .	6-15
6.6.	The experimental $L_{\text{SINR}}$ curve versus secondary data set size for the partial dimension CSM algorithm using 65 DOF. . . . .	6-16
6.7.	The experimental $L_{\text{SINR}}$ curve versus secondary data set size for the partial dimension CSM algorithm using 45 DOF. . . . .	6-17
6.8.	Overlay of the $L_{\text{SINR}}$ curves for reduced dimensions below the dimension of the interference subspace. . . . .	6-18

Figure		Page
6.9.	Singular value and cross spectral metric magnitudes of the known interference plus noise covariance matrix. . . . .	6-20
6.10.	Singular values of the interference plus noise covariance matrix. . . . .	6-21
6.11.	Overlay of the experimental SINR curves for reduced dimensions above the dimension of the interference subspace. . . . .	6-23
6.12.	The CSM ordering of the eigenvectors for a secondary data set size $J$ equal to 323, one less than $MN$ . . . . .	6-26
6.13.	The impact of non-homogeneities in the secondary data on the CSM algorithm. . . . .	6-29
6.14.	The impact of using the GIP Non-Homogeneity detector with the CSM algorithm. . . . .	6-30
6.15.	The impact of using the GIP Non-Homogeneity detector with the CSM algorithm. . . . .	6-31

*List of Tables*

Table		Page
3.1.	Scenario parameters for the clutter example. . . . .	3-32
5.1.	Target parameters used for the GIP non-homogeneity detector example. . .	5-10
A.1.	Properties of Kronecker Products . . . . .	A-1

# *List of Symbols*

Symbol	Page
$\omega_o$ temporal frequency . . . . .	2-2
$\mathbf{k}_o$ spatial frequency vector . . . . .	2-2
$\lambda_o$ nominal wavelength . . . . .	2-2
$c$ speed of light . . . . .	2-2
$f_o$ transmitted frequency . . . . .	2-2
$\hat{\mathbf{k}}$ directional unit vector . . . . .	2-2
$\alpha$ slowness vector . . . . .	2-2
$\mathbf{x}$ position vector . . . . .	2-3
$(\mathbf{k}, \omega)$ wavenumber-frequency space . . . . .	2-3
$\mathbf{k}$ wavenumber vector . . . . .	2-3
$s(\mathbf{x}, t)$ space-time signal . . . . .	2-4
$v_p$ velocity of propagation . . . . .	2-8
$N$ number of antenna elements . . . . .	2-8
$d$ interelement spacing . . . . .	2-9
$r_i(t)$ received signal at $i^{th}$ antenna element . . . . .	2-9
$s(\mathbf{x}_i, t)$ space-time signal sampled at position $\mathbf{x}_i$ . . . . .	2-9
$\text{bf}(t)$ beamformer output signal . . . . .	2-10
$\tau_i$ beamforming delay for $i^{th}$ antenna element . . . . .	2-10
$a$ additional propagation distance from element to element . . . . .	2-10
$W(\mathbf{k} - \omega\alpha_o)$ antenna array pattern . . . . .	2-13
$w$ uniform antenna element weight . . . . .	2-15
$\hat{\mathbf{x}}$ unit vector in the $x$ axis direction . . . . .	3-3
$\mathbf{v}_a$ magnitude and direction of the platform velocity . . . . .	3-3
$\phi$ azimuth angle . . . . .	3-3
$\theta$ elevation angle . . . . .	3-3
$\hat{\mathbf{y}}$ unit vector in the $y$ axis direction . . . . .	3-4

Symbol	Page
$\hat{z}$ unit vector in the $z$ axis direction . . . . .	3-4
$\psi$ random phase shift . . . . .	3-4
$a_t$ transmitted pulse amplitude . . . . .	3-4
$u(t)$ pulse train envelope function . . . . .	3-4
$T_r$ Pulse Repetition Interval . . . . .	3-4
$M$ number of pulses in CPI . . . . .	3-4
$E_u$ fundamental pulse energy . . . . .	3-5
$E_p$ total energy transmitted in the pulse train . . . . .	3-5
$E_t$ transmitted signal energy . . . . .	3-5
$f_t$ Doppler shift of target . . . . .	3-5
$v_t$ target's relative velocity between target and platform . . . . .	3-6
$a_r$ received pulse amplitude . . . . .	3-6
$f_r$ Pulse Repetition Frequency . . . . .	3-6
$\bar{\omega}$ normalized Doppler frequency . . . . .	3-6
$\tau_t$ round trip time to the target . . . . .	3-7
$R_t$ range to target . . . . .	3-7
$\chi(\tau_t, f_t)$ radar ambiguity function . . . . .	3-8
$l$ range gate of interest within CPI . . . . .	3-9
$x_{nm}$ target samples at the $n^{th}$ antenna element and $m^{th}$ pulse . . . . .	3-10
$m$ pulse of interest within CPI . . . . .	3-10
$\mathbf{x}_m$ column vector containing each antenna element sample at a particular range gate $l$ and pulse $m$ . . . . .	3-10
$\mathbf{a}_t(\vartheta_t)$ spatial steering vector . . . . .	3-10
$\mathbf{X}$ matrix of incoming antenna samples . . . . .	3-11
$\mathbf{b}_t(\bar{\omega}_t)$ temporal steering vector . . . . .	3-11
$\otimes$ Kronecker product . . . . .	3-11
$\mathbf{v}_t$ space-time steering vector . . . . .	3-11

Symbol	Page
$\chi_t$ space-time snapshot . . . . .	3-12
$\chi_u$ space-time snapshot for undesired components . . . . .	3-12
$\sigma^2$ the noise power in a single element . . . . .	3-14
$N_o$ noise power spectral density . . . . .	3-14
$B$ bandwidth of the receiver . . . . .	3-14
$\delta$ Kronecker delta . . . . .	3-14
$\chi_n$ spatial snapshot for thermal noise . . . . .	3-14
$\mathbf{R}_n$ noise space-time covariance matrix . . . . .	3-14
$\mathbf{I}$ identity matrix . . . . .	3-14
$B_j$ bandwidth of the jammer . . . . .	3-14
$T_p$ pulse width . . . . .	3-15
$\text{sa}(x)$ sine over argument function . . . . .	3-15
$\alpha_m$ jammer amplitude . . . . .	3-16
$\mathbf{a}_j$ jammer steering vector . . . . .	3-16
$\chi_j$ jammer space-time snapshot . . . . .	3-17
$\xi_j$ Jammer-to-Noise Ratio . . . . .	3-17
$S_j$ radiated power spectral density of jammer . . . . .	3-17
$A_e$ effective area of the receive antenna . . . . .	3-17
$R_j$ range to the jammer . . . . .	3-17
$g(\theta, \phi)$ element power pattern . . . . .	3-18
$L_r$ receiver losses . . . . .	3-18
$J_o$ received jammer power spectral density . . . . .	3-18
$\mathbf{R}_j$ jammer space-time covariance matrix . . . . .	3-18
$P$ number of jammers . . . . .	3-19
$L$ number of unambiguous range bins . . . . .	3-22
$N_c$ number of clutter patches per constant range ring . . . . .	3-22
$R_c$ range to the clutter ring . . . . .	3-22



Symbol	Page
$h_a$ aircraft altitude . . . . .	3-22
$a_e$ effective radius of the Earth . . . . .	3-22
$\psi_c$ grazing angle in clutter model . . . . .	3-23
$R_u$ unambiguous range . . . . .	3-23
$N_r$ number of range ambiguities to the horizon . . . . .	3-24
$\chi_c$ clutter space-time snapshot . . . . .	3-24
$\sigma_c$ RCS due to the area circumscribed by $A_c$ . . . . .	3-24
$A_c$ area of clutter patch . . . . .	3-24
$\gamma$ terrain dependent RCS factor . . . . .	3-25
$\mathbf{R}_c$ clutter space-time covariance matrix . . . . .	3-26
$\beta$ slope of clutter ridge . . . . .	3-28
$F_n$ system noise figure . . . . .	3-36
$G_r$ receiver gain . . . . .	3-36
$k$ Boltzman's constant . . . . .	3-36
$\mathbf{w}$ fully adaptive weight vector . . . . .	3-40
$\mathbf{R}$ interference plus noise covariance matrix (all undesired components) . . . . .	3-40
$z$ output of fully adaptive STAP algorithm . . . . .	3-40
$\mathbf{t}$ two dimensional window/taper . . . . .	3-41
$\odot$ Hadamard product . . . . .	3-41
$\mathbf{t}_b$ temporal window . . . . .	3-41
$\mathbf{t}_a$ spatial window . . . . .	3-41
$\xi_t$ target SNR per element . . . . .	3-42
$P_{\mathbf{w}}(\vartheta, \bar{\omega})$ adapted antenna pattern . . . . .	3-46
$\mathbf{W}$ DFT matrix . . . . .	3-51
$z_m$ output of the FTS algorithm for the $m^{th}$ Doppler bin . . . . .	3-52
$L_{\text{SINR}}$ SINR Loss . . . . .	3-53
$P_{\text{FA}}$ probability of false alarm . . . . .	4-1

Symbol	Page
$P_D$ probability of detection . . . . .	4-2
$\mathbf{s}$ normalized space-time steering vector . . . . .	4-4
$y$ output of the fully adaptive GSC/CSM algorithm . . . . .	4-5
$\mathbf{T}$ transformation matrix . . . . .	4-6
$\mathbf{B}$ blocking matrix . . . . .	4-6
$\mathbf{b}$ noise-subspace data vector . . . . .	4-6
$\mathbf{w}_{GSC}$ optimal weight vector for the GSC adaptive processing model . . . . .	4-8
$d$ desired signal component in CSM/GSC algorithm . . . . .	4-10
$\mathbf{w}_b$ weight vector in fully adaptive GSC/CSM algorithm . . . . .	4-10
$\mathbf{R}_b$ covariance matrix for the data vector $\mathbf{b}$ . . . . .	4-11
$\mathbf{U}$ unitary matrix of eigenvectors . . . . .	4-11
$\mathbf{\Lambda}$ diagonal matrix of eigenvalues . . . . .	4-11
$\mathbf{p}$ principal coordinate process . . . . .	4-12
$\mathbf{R}_p$ covariance matrix of $\mathbf{p}$ . . . . .	4-12
$\mathbf{w}_p$ weight vector for GSC model using full eigenvector projection matrix . . . . .	4-12
$\mathbf{U}$ reduced dimension eigenvector projection matrix . . . . .	4-13
$O$ number of columns used to construct $\mathbf{U}$ . . . . .	4-13
$\mathbf{z}$ output of the dimension reduction operation in CSM model . . . . .	4-13
$\mathbf{R}_z$ CSM model reduced dimension covariance matrix . . . . .	4-15
$\mathbf{\Lambda}_O$ diagonal matrix composed of the eigenvalues corresponding to the eigenvectors used to construct $\mathbf{U}$ . . . . .	4-15
$\mathbf{r}_{sd}$ cross correlation vector between the reduced dimension noise subspace vector, $\mathbf{z}$ , and $d$ . . . . .	4-15
$\mathbf{w}_z$ CSM reduced dimension weight vector . . . . .	4-15
$P$ MMSE . . . . .	4-16
$y_r$ output of the reduced dimension GSC/CSM model . . . . .	4-16
$\mathbf{e}$ error between the reduced dimension weight vector and the full dimension weight vector . . . . .	4-17

Symbol	Page
$\hat{\mathbf{R}}_i$ interference plus noise covariance matrix estimate . . . . .	5-2
$J$ number of secondary data vectors to be used in estimating $\hat{\mathbf{R}}$ . . . . .	5-2
$N_{\text{DOF}}$ number of Degrees Of Freedom . . . . .	5-3
$\chi_{ci}$ homogeneous clutter component of space-time snapshot . . . . .	5-5
$\chi_{ji}$ white noise jamming component of space-time snapshot . . . . .	5-5
$\chi_{ni}$ uncorrelated noise component of space-time snapshot . . . . .	5-5
$\hat{\mathbf{R}}_s$ interference plus noise covariance matrix estimate based on the entire secondary data set . . . . .	5-5
$g_i$ GIP for the $i^{\text{th}}$ secondary data vector . . . . .	5-5
$\bar{g}$ average of all the GIPs for the secondary data set . . . . .	5-6
$\text{SINR}_{\text{CSM full}}$ fully adaptive CSM algorithm output SINR . . . . .	6-3
$\text{SINR}_{\text{CSM part}}$ partially adaptive CSM algorithm output SINR . . . . .	6-7
$\mathbf{Q}$ Cholesky decomposition . . . . .	6-10
$\tilde{\chi}$ artificially generated space-time snapshot . . . . .	6-10
$\tilde{\mathbf{R}}$ covariance matrix estimate of artificially generated space-time snapshot . . . . .	6-10

## *List of Abbreviations*

Abbreviation	Page
ECCM Electronic Counter-Countermeasure . . . . .	1-1
STAP Space-Time Adaptive Processing . . . . .	1-1
MCARM Multi-Channel Airborne Radar Management . . . . .	1-3
PD Pulse Doppler . . . . .	3-4
PRI Pulse Repetition Interval . . . . .	3-4
CPI Coherent Processing Interval . . . . .	3-4
RCS Radar Cross Section . . . . .	3-6
PRF Pulse Repetition Frequency . . . . .	3-6
JNR Jammer-to-Noise Ratio . . . . .	3-17
CNR Clutter-to-Noise Ratio . . . . .	3-24
SINR Signal-to-Interference-plus-Noise-Ratio . . . . .	3-40
FTS Factored Time-Space . . . . .	3-50
DOF Degrees Of Freedom . . . . .	3-50
DFT Discrete Fourier Transform . . . . .	3-50
SNR Signal-to-Noise-Ratio . . . . .	4-2
GSC Generalized Sidelobe Canceler . . . . .	4-5
MMSE Minimum Mean Square Error . . . . .	4-16
CSM Cross Spectral Metric . . . . .	4-27
MLE Maximum Likelihood Estimate . . . . .	5-2
SMI Sample Matrix Inversion . . . . .	5-3
iid Independent and Identically Distributed . . . . .	5-4
GIP Generalized Inner Product . . . . .	5-5
CFAR Constant False Alarm Rate . . . . .	5-13
SVD Singular Value Decomposition . . . . .	6-19

*Abstract*

One of the primary problems with the application of Space-Time Adaptive Processing (STAP) techniques is secondary data support for the interference plus noise covariance matrix estimate. This estimate is obtained using a Maximum Likelihood Estimator (MLE). As always in the case of a statistical estimator, the larger the number of points, or secondary data support, used in the calculation the more accurate the estimate.

Reed [15] has shown the required secondary data support in this estimate to achieve performance within 3 dB of the optimal signal-to-interference-plus-noise ratio (SINR) is approximately two times the Degrees Of Freedom (DOF). Hence, reducing the DOF used in the adaptive algorithm also reduces required secondary data support. Reed developed this rule for Sample Matrix Inversion (SMI) STAP techniques.

A concern arises when applying this rule to a class of newer STAP methods designed to reduce the DOF. An example of one of these newer methods is the Cross Spectral Metric (CSM) algorithm [6]. Since the CSM algorithm does not fall under the SMI classification, application of Reed's rule for required secondary data support is brought into question. In an effort to answer this question, extensive Monte Carlo experiments were performed. These experiments have resulted in a number of interesting conclusions.

When using full dimensionality (all DOF), the CSM algorithm performs as predicted by Reed's rule. However, when reducing the DOF the required secondary data support does not follow Reed's prediction. Optimal SINR performance is obtained only when the DOF are equal to the dimension of the interference subspace. Furthermore, there is a threshold below the dimension of the interference subspace where reducing the DOF does not gain any reduction in secondary data support.

A second goal of the thesis determines the impact of non-homogeneities within the secondary data on the CSM algorithm. The Generalized Inner Product (GIP) detection scheme is then used to excise these non-homogeneities from the secondary data. The CSM algorithm was found to be susceptible to non-homogeneities. The use of the GIP successfully negated the impact on this algorithm.

# Secondary Data Support and Non-Homogeneities in Space-Time Adaptive Processing

## *I. Introduction*

### *1.1 Background*

Clutter and jamming are serious problems for airborne radar engineers. Historically, the elimination of clutter has been accomplished through very low antenna sidelobes, an expensive and difficult proposition in the case of airborne radars. Furthermore, Electronic Counter-Countermeasure (ECCM) techniques currently implemented in radars are reactionary, meaning they are relegated to respond defensively instead of offensively. Space-Time Adaptive Processing (STAP) promises an inexpensive alternative capable of filtering clutter and allowing an offensive approach to jamming by removing it before the radar processes the return. STAP refers to adaptive filtering of antenna array inputs in both the spatial and temporal domains to suppress the effects of clutter and jamming. Current research shows it is an effective approach that significantly improves the performance of an airborne radar.

STAP is a digital filtering technique applicable to any antenna array and holds the potential to significantly improve radar performance at minimal cost. The mission of Rome Laboratory (the sponsor of this research project) is to provide cost effective solutions to the technological problems of the Air Force. STAP fits perfectly into their mission. This research also supports the broader Air Force goal to bring effective weapon platforms to the battlefield while minimizing cost.

## 1.2 Objective

Although there are a variety of STAP techniques available, each with their own strengths and weaknesses, all of these techniques rely on the statistics of the interference and clutter, or the interference covariance matrix  $\mathbf{R}$ . The widely varying clutter and interference characteristics of an airborne environment effectively prohibits prediction of these statistics. The interference covariance structure must be estimated from secondary data available within the radar's Coherent Processing Interval (CPI).

This thesis focuses on a method of STAP just recently introduced. The Cross Spectral Metric (CSM) algorithm [6] results in an adaptive choice of the Degrees of Freedom (DOF) that result in optimal SINR. The cross-spectral metric approach was developed by Goldstein and Reed [8] as an alternative to the method of principle components. Chapter IV discusses these approaches in more detail. Because of the methods used to determine the optimum DOF, there is some question as to the secondary data support required to effectively estimate the covariance structure of the environment. This research effort attempts to answer this question.

Once the required secondary data support has been determined for the CSM algorithm, the thesis progresses to another problem prevalent in all STAP algorithms. If a target or some other non-homogeneity is present within the secondary data, the covariance matrix estimate is distorted. Depending on the severity of the non-homogeneity in the secondary data, this estimate can distort the antenna pattern produced by the STAP algorithm. This research project illustrates the impact of secondary data non-homogeneities on the CSM algorithm.

In an effort to alleviate the effect of these non-homogeneities, Melvin [14] has proposed a non-homogeneity detection scheme. Detection is performed through the use of a Generalized Inner Product (GIP). He has shown the improvements gained through its use for the reduced dimension Factored Time-Space (FTS) approach [14]. As a final objective, this research applies the GIP detection scheme to the CSM algorithm. MATLAB<sup>®</sup> is used to explore these alternatives.

### *1.3 Material and Equipment*

All simulations were developed and performed using MATLAB<sup>®</sup> versions 4.2c and 5.0, a high performance numeric computation and visualization software optimized for matrix operations. The software was operated on Sun Workstations provided by the Air Force Institute of Technology.

### *1.4 Support*

Capt William L. Melvin, Rome Laboratory, Griffis Industrial Park, NY served as the point of contact for this thesis. Many thanks go to Rome Laboratory for allowing the use of Multi-Channel Airborne Radar Management (MCARM) data. This data represents real world collections from an antenna array using the MCARM airborne radar platform. Readers interested in more details should reference the final technical report [18].

### *1.5 Organization*

This document serves to evaluate required secondary data support for the CSM algorithm and the impact of non-homogeneities among secondary data used for interference plus noise covariance matrix estimation. In order to present a coherent approach, the fundamentals of beamforming and STAP are first explored. Chapter II delves into the signal processing viewpoint of antenna array beamforming. The basic concepts used in the STAP development are derived and discussed there.

Chapter III develops the basics of STAP in parallel with Ward [21]. Both fully adaptive and partially adaptive algorithms are introduced. Because of the computational burden associated with fully adaptive STAP, this algorithm is beyond current computing capabilities for real time operation in an airborne platform. Although there are many reduced dimension STAP algorithms available, this chapter only introduces the Factored Time-Space (FTS) method. The FTS method is simple to implement and effectively introduces the advantages of reduced dimensionality with respect to the computational load of STAP.



Chapter IV is devoted to developing an understanding of Goldstein's [8] Cross Spectral Metric (CSM) algorithm. The CSM is introduced as a means of selecting an optimum dimension reduction matrix used to reduce the DOF and indirectly the secondary data support.

The final set of background information necessary is a complete discussion of the Generalized Inner Product (GIP) non-homogeneity detector [14]. Chapter V not only introduces this detection scheme but also presents a comprehensive example of its application to the FTS algorithm using MCARM data.

The analysis of the CSM algorithm is contained in Chapter VI. There are two major questions answered in this thesis. The first is in regard to sample support required for the CSM model. Monte Carlo simulation is employed to answer this question. The second question of interest involves application of the GIP non-homogeneity detector to the CSM model. The goal is to determine if performance is improved.

## II. General Beamforming Concepts

### 2.1 Introduction

People are bombarded with plane waves every day, but the average person doesn't think of them in this manner. The greeting from a friend in the distance is carried on a plane wave, an acoustic wave in this simple example. The visual image of that friend is carried on plane waves from a relatively small portion of the electromagnetic spectrum. Extending these interpretations, radar and sonar become simple extensions of the basic human sensory organs. They give us the ability to make use of portions of the electromagnetic spectrum beyond our own physiological limitations.

This first chapter introduces a mathematical model for these waves from basic electromagnetic principles. It then highlights the functional similarities between the expressions developed using this model and an expression for the multidimensional Fourier transform. This similarity shows the applicability of multidimensional signal processing in the analysis of these signals, referred to as *space-time signals*. Beamforming and Space-Time Adaptive Processing (STAP) are found to be ideal methods to "analyze" these signals and extract the information of interest.

This chapter follows closely the development by Dudgeon and Mersereau [5, pages 289–303]. The chapter lays the groundwork for the later concepts of STAP.

### 2.2 Foundations

The mathematical representation for a propagating plane wave from electromagnetic fundamentals is in terms of a complex exponential,

$$\begin{aligned} s(\mathbf{x}, t) &= e^{j(\omega_o t - \mathbf{k}_o \cdot \mathbf{x})} \\ &= e^{j\omega_o(t - \frac{\mathbf{k}_o \cdot \mathbf{x}}{\omega_o})}, \end{aligned} \tag{2.1}$$

where the variable  $\omega_o$  represents the temporal frequency of the wave and the vector  $\mathbf{k}_o$  designates the direction of the propagating wave and its spatial frequency. The dot product in the expression represents the inner product between the two vectors and can also be written as

$$\mathbf{k}_o^T \mathbf{x}, \quad (2.2)$$

where the superscript  $T$  represents the transpose operation. The spatial frequency vector can be described by

$$\mathbf{k}_o = \frac{2\pi}{\lambda_o} \hat{\mathbf{k}}, \quad (2.3)$$

where  $\lambda_o$  is the nominal wavelength of the propagating wave (equal to  $\frac{c}{f_o}$ , where  $c$  is the velocity of propagation and  $f_o$  is the transmitted frequency) and  $\hat{\mathbf{k}}$  is a unit vector in the direction of the propagating wave. The form of Eqn. (2.1) results in the definition of a new parameter  $\alpha$ ,

$$\alpha_o = \frac{\mathbf{k}_o}{\omega_o} \left( \frac{\text{sec}}{\text{meter}} \right), \quad (2.4)$$

where the units of this new quantity are the inverse of velocity. Hence, it is justifiably called the *slowness vector*. The  $_o$  subscript depicts the slowness vector is in the direction of interest, or reference direction. Generically, the slowness vector is written without the subscript. The plane wave expression describing a signal propagating in the direction of interest written in terms of the slowness vector then becomes

$$s(\mathbf{x}, t) = e^{j\omega_o(t - \alpha_o \cdot \mathbf{x})}. \quad (2.5)$$

The slowness vector  $\alpha_o$  not only describes the speed of the wave,  $1/|\alpha_o|$ , but also it's direction. The plane wave expression can be written generically as a function of the speed of the wave and

direction by dropping the subscript on  $\alpha$  and  $\omega$ ,

$$s(\mathbf{x}, t) = e^{j\omega(t - \alpha \cdot \mathbf{x})}. \quad (2.6)$$

Examine the standard Fourier transform definition as shown by

$$\begin{aligned} S(\Omega) &= \int_{-\infty}^{\infty} s(\mathbf{t}) e^{-j(\Omega \cdot \mathbf{t})} d\mathbf{t} \\ &= \int_{-\infty}^{\infty} s(t_1, t_2, \dots, t_n) e^{-j(\Omega_1 t_1 + \Omega_2 t_2 + \dots + \Omega_n t_n)} dt_1 dt_2 \dots dt_n. \end{aligned} \quad (2.7)$$

The expression illustrates a conflict with the sign in the exponential and our knowledge of propagating waves, i.e. this representation does not match the sign convention normally used to represent the direction of a propagating plane wave.

To match this convention, the Fourier transform is redefined. The 4-D case (using the propagating plane wave as the argument) becomes

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k} \cdot \mathbf{x})} d\mathbf{x} dt. \quad (2.8)$$

Notice the sign is only changed on the position term,  $\mathbf{x}$ . The equation now correctly reflects the direction of the plane wave from basic electromagnetics.

**2.2.1 Wavenumber-Frequency Space.** The  $(\mathbf{k}, \omega)$  space obtained from the Fourier transform is referred to as *wavenumber-frequency space*. The vector  $\mathbf{k}$  is the *wavenumber vector*. This vector represents the spatial frequency of the wave, while  $\omega$  represents the familiar temporal frequency. Applying the 4-D Fourier transform to the propagating plane wave of Eqn. (2.1) uncovers an important property of the frequency spectrum of plane waves. The sequence of operations

uncovering this property is

$$\begin{aligned}
S(\mathbf{k}, \omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{j(\omega_o t - \mathbf{k}_o \cdot \mathbf{x})} e^{-j(\omega t - \mathbf{k} \cdot \mathbf{x})} d\mathbf{x} dt \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-j(\omega - \omega_o)t} dt e^{-j(\mathbf{k}_o - \mathbf{k}) \cdot \mathbf{x}} d\mathbf{x} \\
&= \int_{-\infty}^{\infty} \delta(\omega - \omega_o) e^{-j(\mathbf{k}_o - \mathbf{k}) \cdot \mathbf{x}} d\mathbf{x} \\
&= \delta(\omega - \omega_o) \int_{-\infty}^{\infty} e^{-j(\mathbf{k}_o - \mathbf{k}) \cdot \mathbf{x}} d\mathbf{x} \\
&= \delta(\omega - \omega_o) \delta(\mathbf{k}_o - \mathbf{k}).
\end{aligned} \tag{2.9}$$

This is a four dimensional impulse at  $\mathbf{k} = \mathbf{k}_o$  and  $\omega = \omega_o$ . Therefore, every point in  $(\mathbf{k}, \omega)$  space corresponds to a plane wave in  $(\mathbf{x}, t)$  space with a particular direction and frequency. For this reason, antenna beamforming lends itself directly to multidimensional signal processing. Designing a bandpass filter in wavenumber-frequency space transforms back to space-time as an antenna beam passing only the propagating waves with a particular direction and frequency.

**2.2.2 Space-Time Signal Decomposition.** Another important concept involves the decomposition of a signal in  $(\mathbf{x}, t)$  space. Any space-time signal  $s(\mathbf{x}, t)$  can be decomposed into a superposition of propagating plane waves. This is shown by the 4-D inverse Fourier transform,

$$s(\mathbf{x}, t) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) \underbrace{e^{j(\omega t - \mathbf{k} \cdot \mathbf{x})}}_{\text{Plane Wave}} d\mathbf{k} d\omega, \tag{2.10}$$

where  $S(\mathbf{k}, \omega)$  is the 4-D Fourier transform of the space-time signal  $s(\mathbf{x}, t)$ . Notice the exponential in the expression exactly resembles the expression for the propagating plane wave given in Eqn. (2.1).  $S(\mathbf{k}, \omega)$  is viewed as the complex magnitude of each individual plane wave. The integrals basically sum all of the individual propagating plane waves in each direction to form the space-time signal,  $s(\mathbf{x}, t)$ .

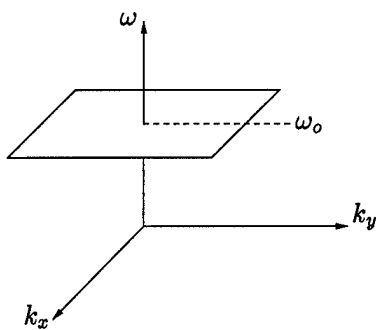


Figure 2.1 Wavenumber-Frequency space representation of space-time signals with the same frequency.

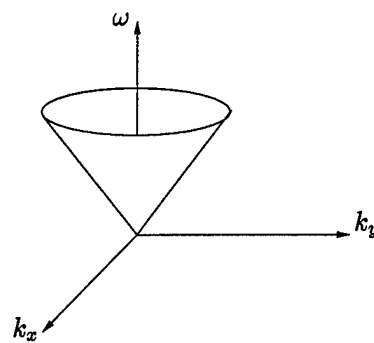


Figure 2.2 Wavenumber-Frequency space representation of space-time signals with the same velocity of propagation.

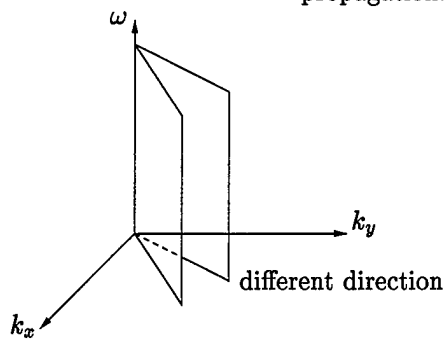


Figure 2.3 Wavenumber-Frequency space representation of space-time signals with the same direction of propagation.

**2.2.3 Wavenumber-Frequency Space Translation.** Using the fundamental ideas just explained, the plots shown in Figs. 2.1 through 2.3 help illustrate how the  $(\mathbf{k}, \omega)$  space relates back to  $(\mathbf{x}, t)$  space. These figures ignore spatial frequency in the  $z$  direction ( $k_z$  from the wavenumber vector) in order to allow representation in three dimensions.

To see how these plots are related, examine the relationship between velocity of propagation, temporal frequency, and spatial frequency. Looking at the units is the easiest way to understand the relationship between these three quantities,

$$v_p = \frac{\omega}{|\mathbf{k}|} \left( \frac{\text{meters}}{\text{sec}} \right). \quad (2.11)$$

Holding the temporal frequency constant without any constraint on the velocity of propagation or direction of the wave results in a horizontal plane in wavenumber-frequency space. This is shown in Fig. 2.1. Forcing the velocity of propagation to be constant as in Fig. 2.2 results in a cone. As the temporal frequency  $\omega$  increases, the magnitude of the spatial frequency must also increase at the same rate in order for the velocity of propagation to be constant. Hence, the area of wavenumber-frequency space described by this relationship is the surface of a cone.

The final relationship of interest is signals propagating in the same direction. These translate to the space shown in Fig. 2.3. The direction of the signal is described by the wavenumber/spatial frequency vector  $\mathbf{k}$ . The direction of the vector is the same as long as its individual components are all operated on in a like manner. Any increase in  $k_x$  must have a corresponding increase in  $k_y$  (the ratio remains constant). Therefore, one direction of propagation is described by a vertical plane (no temporal frequency restriction) in wavenumber-frequency space. The plot shown illustrates two different directions.

The intersections of these plots give a visual interpretation to the signal processing aspect of beamforming. The object of beamforming is to eliminate undesired signals and focus only on signals of interest. Generally the desired signal is one of a particular direction, velocity of propagation, and/or frequency. Signals with the same velocity of propagation and direction lie along a line formed by the cone and vertical planes shown in Fig. 2.2 and 2.3. This line can be interpreted as a bandpass filter as shown in Fig. 2.4. By also choosing a frequency, this intersection line collapses down to a point. In radar, frequency is generally of no concern since a typical radar system is trying to receive the "echos" from its own transmission. The existing roll-off of the amplifiers is usually adequate to remove undesired frequency components.

In summary, let us translate some of the basic ideas to practical concepts. The position vector  $\mathbf{x}$  represents a three dimensional array of antennas used to spatially sample the signal. The signal applied to each antenna element (used synonymously with receiver) is the spatially sampled

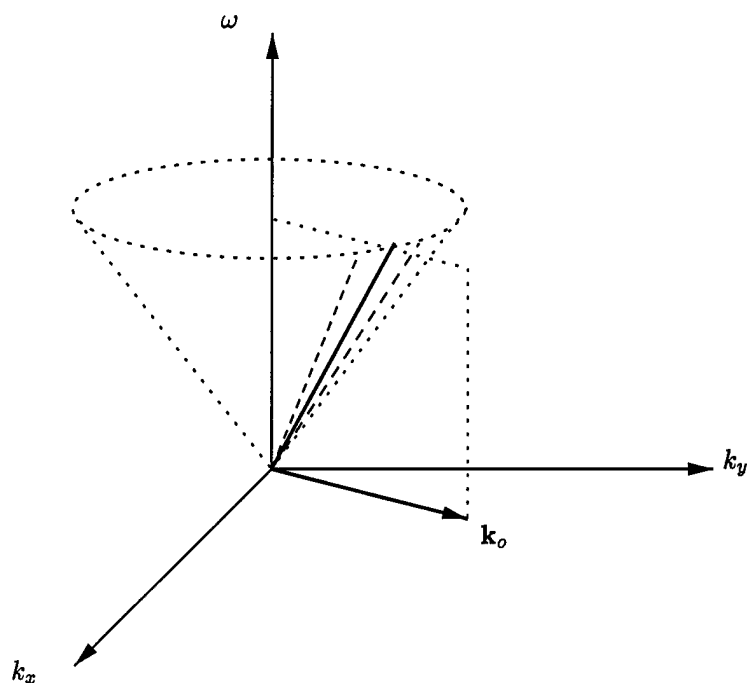


Figure 2.4 The intersection of the constant velocity of propagation cone and the directional plane in wavenumber-frequency space.

signal  $s(\mathbf{x}_i, t)$ , where  $i$  denotes the antenna element. Temporal (time) sampling provides the final dimension. Superposition allows the space-time signal to be broken down into a sum of individual plane wave components, each of a particular frequency and direction. The appropriate bandpass filter removes undesired signal components.

### 2.3 Weighted Delay and Sum Beamformer

The objective of beamforming is to isolate signals propagating in the same direction. Figure 2.5 graphically illustrates the idea of forming a beam or *passband* in the direction of the target. Undesired signals are attenuated by the roll-off of the antenna pattern. Ideally, the beamformer passes the desired target (direction and speed represented by the slowness vector  $\alpha_o$ ) and totally rejects the undesired signals.



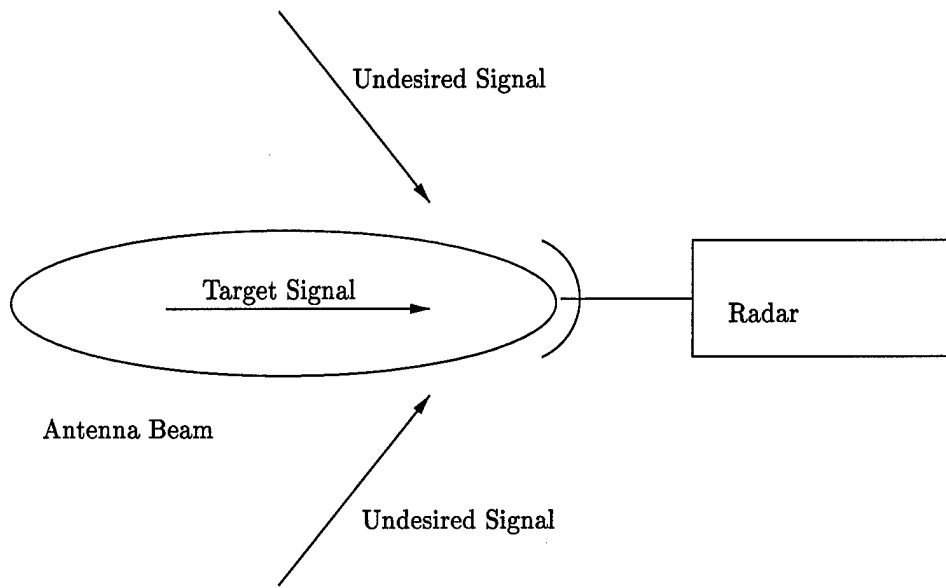


Figure 2.5 A block diagram of the signal environment.

Assuming all waves propagate at the same speed, the beamformer becomes the intersection of the cone in Fig. 2.2 and the vertical plane in Fig. 2.3. This describes signals of any frequency constrained to a particular direction and velocity of propagation. The assumption of a constant velocity of propagation  $v_p$  is valid in the discussion of radar since the wave is propagating in free space, generally assumed a homogeneous medium. The wavenumber-frequency space diagram depicting this is shown in Fig. 2.4. The dashed lines on each side of the intersection line represent the passband of the *beamforming* filter.

As mentioned at the end of the previous section, the directional vector  $\mathbf{x}$  is interpreted as the location of the antenna elements. For an antenna consisting of  $N$  elements, each element position is described by a separate vector  $\mathbf{x}_i$ . The diagram in Fig. 2.6 shows the antenna elements lined up in the  $\hat{\mathbf{y}}$  direction, the convention to be used in this chapter. Assuming these elements are spaced equal distances apart, the position vector describing each element location becomes

$$\mathbf{x}_i = id\hat{\mathbf{y}} \text{ for } i = 0 : N - 1, \quad (2.12)$$

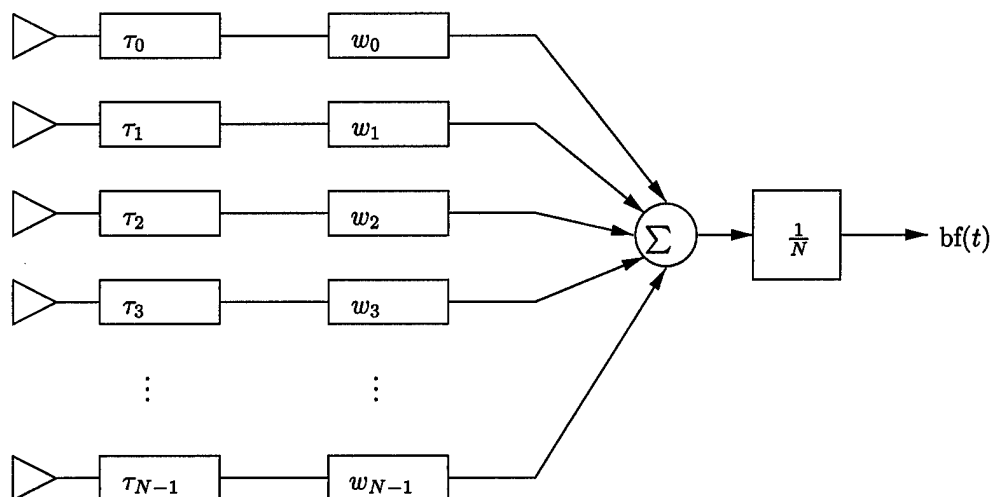


Figure 2.6 Block diagram of the weighted delay and sum beamformer. The output of the beamformer as a function of time is designated  $bf(t)$ .

where  $d$  is the interelement spacing. The reference element is assigned the zero position in the coordinate system, referred to as element zero. Using this notation, the received signal (denoted  $r_i(t)$ ) from each element becomes  $s(\mathbf{x}_i, t)$ , the space-time signal sampled at position  $\mathbf{x}_i$ .

The block diagram in Fig. 2.6 also shows a weight applied to each received signal. At this point in the derivation the weight is taken to be a real valued constant for each element, although not constrained to the same value for every element. Complex weights are time delays and hence are constrained to the delay blocks. Without any delays on the elements, the beam is perpendicular to the face of the antenna array (the  $\hat{\mathbf{x}}$  direction). However, placing individually selectable complex exponential weights (time delays) after each antenna element steers the beam. The sum and  $1/N$  blocks after the weight and delay blocks serve to average the signal received from each element. Called the *weighted delay and sum* beamformer, this beamformer's name is a direct derivative of its elements of operation.

**2.3.1 Beam Steering Through Time Delays.** The delay concept is simple. A plane wave propagating in some direction other than perpendicular to the array arrives at different times on each element. Applying the appropriate delay in each element's path before summing all the

signals together allows a reference point on the signal to be in phase at the sum block. Hence, it adds constructively and produces a large output from the beamformer. However, a plane wave propagating in a different direction experiences phase differences when it reaches the sum block. A weak beamformer output signal  $bf(t)$  results from the destructive addition. Delays,  $\tau_i$ , given by

$$\tau_i = -\alpha_o \cdot \mathbf{x}_i \quad (2.13)$$

pass waves in the direction of the slowness vector [5, page 294].

For example, examine the diagram in Fig. 2.7. Drawing a reference line perpendicular to the direction of the propagating wave and based at element zero shows the increase in distance a reference point on the wave travels to reach element one. The time to travel this distance is the negative delay required in element one. Calculating this delay is simply a matter of dividing the distance traveled by the velocity of propagation. In this simple example, all of the elements are lined up vertically ( $\hat{y}$  direction) and equally spaced. Therefore, once the required delay for element one is found the rest of the delays are integer multiples of it.

Given  $d$  as the distance between the elements, the additional travel distance  $a$  (see Fig. 2.7) is

$$a = d \cos(\theta). \quad (2.14)$$

The velocity of propagation is the temporal frequency of the wave multiplied by the wavelength,

$$c = f_o \lambda. \quad (2.15)$$

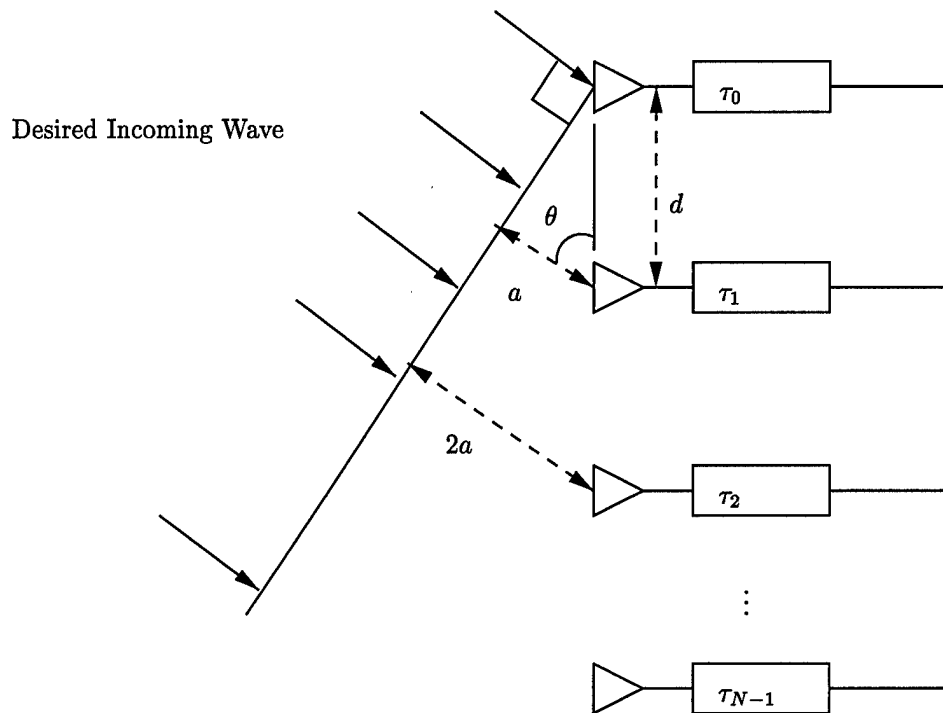


Figure 2.7 Calculation of the delay required in each element to steer the beam.

Therefore, the delay required in element one for the reference point to arrive simultaneously at the sum block from elements zero and one is

$$\begin{aligned}
 \tau &= -\frac{a}{c} \\
 &= -\frac{d \cos(\theta)}{f_o \lambda} \\
 &= -\frac{2\pi d \cos(\theta)}{\lambda \cdot 2\pi f_o} \\
 &= -\frac{d \cos(\theta)}{\omega_o} \underbrace{\frac{2\pi}{\lambda}}_{\text{Spatial Frequency}}
 \end{aligned}$$

Recognizing the spatial frequency term, the equation becomes

$$\begin{aligned}
 \tau &= -\frac{|\mathbf{k}_o|}{\omega_o} d \cos(\theta) \\
 &= -|\boldsymbol{\alpha}_o| d \cos(\theta) \\
 &= -\boldsymbol{\alpha}_o \cdot \mathbf{x}_i,
 \end{aligned} \tag{2.16}$$

where the final step shows the dot product between the two vectors (the same as a vector multiplication) and gives the expression provided by Dudgeon and Mersereau [5, page 294]. Examining the element position vector ( $i = 1$  for element one) in this example,

$$\mathbf{x}_i = [0 \quad id \quad 0]^T, \tag{2.17}$$

shows how the dot product of Eqn. (2.16) collapses to the expression on the previous line.

Obviously, a negative time delay is actually an advance and hence not possible in the real world. Adding the absolute value of  $\tau_{N-1}$  to all of the delay blocks produces the more conventional notion of a positive time delay. In this case,  $\tau_{N-1}$  becomes the delay block equal to zero instead of  $\tau_0$ .

**2.3.2 Antenna Array Pattern.** Examination of the weighted delay and sum beamformer block diagram in Fig. 2.6 allows a derivation of a mathematical expression for the output. The beamformer output is expressed as

$$\text{bf}(t) = \frac{1}{N} \sum_{i=0}^{N-1} w_i r_i(t - \tau_i), \tag{2.18}$$

where  $r_i(t - \tau_i)$  is the received signal sampled at a particular position  $\mathbf{x}_i$ . Assuming the incoming signal is a single plane wave, then (using the plane wave expression from Eqn. (2.5) and the delay

expression from Eqn. (2.13)) the beamformer output becomes

$$\begin{aligned}
 \text{bf}(t) &= \frac{1}{N} \sum_{i=0}^{N-1} w_i s(\mathbf{x}_i, t + \alpha_o \cdot \mathbf{x}_i) \\
 &= \frac{1}{N} \sum_{i=0}^{N-1} w_i e^{j\omega(t + \alpha_o \cdot \mathbf{x}_i - \alpha \cdot \mathbf{x}_i)} \\
 &= \frac{1}{N} \sum_{i=0}^{N-1} w_i e^{j\omega(\alpha_o - \alpha) \cdot \mathbf{x}_i} e^{j\omega t} \\
 &= \frac{1}{N} \sum_{i=0}^{N-1} \underbrace{w_i e^{-j(\mathbf{k} - \omega \alpha_o) \cdot \mathbf{x}_i}}_{W(\mathbf{k} - \omega \alpha_o)} e^{j\omega t}, \tag{2.19}
 \end{aligned}$$

where, the underbraced term is defined as  $W(\mathbf{k} - \omega \alpha_o)$ , the *antenna array pattern* [5].

The above development of the antenna array pattern assumed the space-time signal consisted of only one propagating wave. However, we have already shown the space-time signal is composed of many plane waves (see Eqn. (2.10)). Using this more general case, the received signal becomes Eqn. (2.10) evaluated at  $\mathbf{x}_i$  and the necessary beam steering delay from Eqn. (2.13),

$$\begin{aligned}
 r_i(t - \tau_i) &= s(\mathbf{x}_i, t - \tau_i) \\
 &= \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) e^{-j(\mathbf{k} - \omega \alpha_o) \cdot \mathbf{x}_i} e^{j\omega t} d\mathbf{k} d\omega. \tag{2.20}
 \end{aligned}$$

Substituting this equation into the generic expression for the delay and sum beamformer (Eqn. (2.18)) results in

$$\text{bf}(t) = \frac{1}{N} \sum_{i=0}^{N-1} w_i \left\{ \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) e^{-j(\mathbf{k} - \omega \alpha_o) \cdot \mathbf{x}_i} e^{j\omega t} d\mathbf{k} d\omega \right\}. \tag{2.21}$$

Rearranging the order of the summation and integral highlights the expression for the antenna array pattern,

$$\begin{aligned} \text{bf}(t) &= \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) \left\{ \frac{1}{N} \sum_{i=0}^{N-1} w_i e^{-j(\mathbf{k} - \omega \boldsymbol{\alpha}_o) \cdot \mathbf{x}_i} \right\} e^{j\omega t} d\mathbf{k} d\omega \\ &= \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) W(\mathbf{k} - \omega \boldsymbol{\alpha}_o) e^{j\omega t} d\mathbf{k} d\omega. \end{aligned} \quad (2.22)$$

Examine the response of the weighted delay and sum beamformer for a space-time signal with all components propagating in the same direction. The space-time signal becomes

$$s(\mathbf{x}, t) = v(t - \boldsymbol{\alpha} \cdot \mathbf{x}) \quad (2.23)$$

for the signal  $v(t)$ , where the component  $-\boldsymbol{\alpha} \cdot \mathbf{x}$  is the time delay from Eqn. (2.13). As before,  $\boldsymbol{\alpha}$  represents the direction of the incoming signal and  $\boldsymbol{\alpha}_o$  represents the steered direction of the beam. The Fourier transform of the incoming signal (as defined to correctly reflect the direction of a plane wave) involves using the time shift property,

$$\begin{aligned} S(\mathbf{k}, \omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k} \cdot \mathbf{x})} d\mathbf{x} dt \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(t - \boldsymbol{\alpha} \cdot \mathbf{x}) e^{-j(\omega t - \mathbf{k} \cdot \mathbf{x})} d\mathbf{x} dt \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} v(t - \boldsymbol{\alpha} \cdot \mathbf{x}) e^{-j\omega t} e^{j\mathbf{k} \cdot \mathbf{x}} dt d\mathbf{x} \\ &= V(\omega) \int_{-\infty}^{\infty} e^{-j(\omega \boldsymbol{\alpha} - \mathbf{k}) \cdot \mathbf{x}} d\mathbf{x} \\ &= V(\omega) \delta(\omega \boldsymbol{\alpha} - \mathbf{k}). \end{aligned} \quad (2.24)$$

This shows the  $(\mathbf{k}, \omega)$  space of a propagating wave is a single point.

Expressing the beamformer output as a function of the space-time signal with all components propagating in the same direction shows the effect of the antenna array pattern. The sifting property

of the delta function greatly simplifies the mathematics involved in reducing the expression,

$$\begin{aligned} \text{bf}(t) &= \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V(\omega) \delta(\omega \alpha - \mathbf{k}) W(\mathbf{k} - \omega \alpha_o) e^{j\omega t} d\mathbf{k} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} V(\omega) W(\omega(\alpha - \alpha_o)) e^{j\omega t} d\omega. \end{aligned} \quad (2.25)$$

If the steered beam direction is the same as the direction of the incoming signal,  $\alpha = \alpha_o$ , then the above integral collapses to the inverse Fourier transform of the original signal,  $V(\omega)$ , multiplied by a constant,

$$\begin{aligned} \text{bf}(t) &= \frac{W(0)}{2\pi} \int_{-\infty}^{\infty} V(\omega) e^{j\omega t} d\omega \\ &= W(0)v(t). \end{aligned} \quad (2.26)$$

As expected, the signal is undistorted. If  $\alpha \neq \alpha_o$ , then the antenna array pattern attenuates the signal. Therefore,  $W(\mathbf{k} - \omega \alpha_o)$  can be viewed as the antenna pattern.

**2.3.3 Linear Antenna Array Example.** As an example of the computation of an antenna pattern for a weighted delay and sum beamformer, consider a linear antenna array with  $N$  elements aligned vertically ( $\hat{\mathbf{y}}$  direction). Given the weights  $w$  are uniform and real valued for all of the antenna elements, calculate the antenna pattern. Assume the antenna elements are equally spaced a distance  $d$  apart and the beam is steered toward the direction  $\mathbf{k}_o$ .



First, we'll determine the delays necessary at each element to steer the beam in this direction.

Referencing Eqn. (2.13) gives

$$\begin{aligned}
 \tau_i &= -\alpha_o \cdot \mathbf{x}_i \\
 &= -\omega_o \mathbf{k}_o \cdot \mathbf{x}_i \\
 &= -\omega_o [k_{ox} \ k_{oy} \ k_{oz}] [0 \ id \ 0]^T \\
 &= -id\omega_o k_{oy} = -id\alpha_{oy} \quad \text{for } i = 0 : N - 1.
 \end{aligned}$$

Finding a closed expression for the antenna array pattern is merely a matter of manipulation.

The antenna pattern for the steered antenna beam given  $\omega = \omega_o$  such that  $\omega\alpha_o = \omega_o\alpha_o = \mathbf{k}_o$  is

$$W(\mathbf{k} - \mathbf{k}_o) = \frac{1}{N} \sum_{i=1}^{N-1} w_i e^{-j(\mathbf{k} - \mathbf{k}_o) \cdot \mathbf{x}_i}.$$

Substituting the values in the exponential argument and simplifying results in

$$\begin{aligned}
 W(\mathbf{k} - \mathbf{k}_o) &= \frac{w}{N} \sum_{i=1}^{N-1} e^{-j(k_y - k_{oy})id} \\
 &= \frac{w}{N} \frac{1 - (e^{-j(k_y - k_{oy})d})^N}{1 - e^{-j(k_y - k_{oy})d}} \\
 &= \frac{w}{N} \frac{1 - e^{-j(k_y - k_{oy})dN}}{1 - e^{-j(k_y - k_{oy})d}} \\
 &= \frac{w}{N} \frac{e^{-j(k_y - k_{oy})dN/2} e^{j(k_y - k_{oy})dN/2} - e^{-j(k_y - k_{oy})dN/2} e^{-j(k_y - k_{oy})dN/2}}{e^{-j(k_y - k_{oy})d/2} e^{j(k_y - k_{oy})d/2} - e^{-j(k_y - k_{oy})d/2} e^{-j(k_y - k_{oy})d/2}} \frac{2j}{2j} \\
 &= \frac{w}{N} \frac{\sin((k_y - k_{oy})dN/2)}{\sin((k_y - k_{oy})d/2)} e^{-j(k_y - k_{oy})d(N-1)/2}.
 \end{aligned}$$

A plot of the magnitude for this beamformer (with  $w = 1$  and  $\mathbf{k}_o = \mathbf{0}$ ) is shown in Fig. 2.8.

Examining the arguments of the sine terms shows the pattern is periodic in  $2\pi/d$  radians/meter.

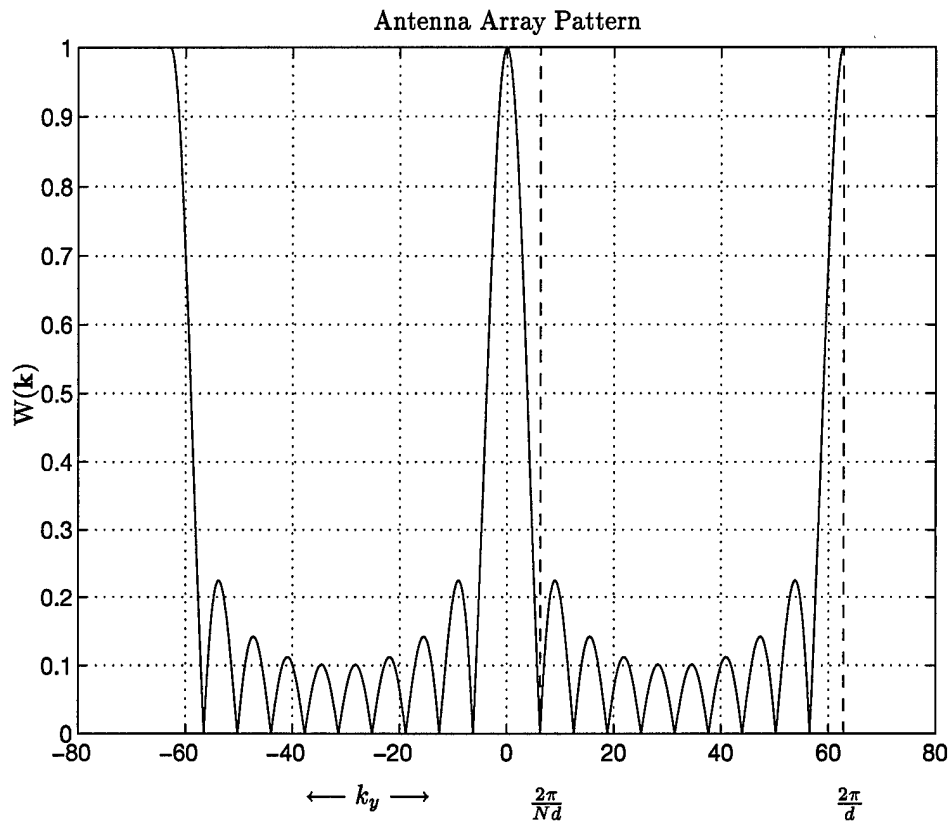


Figure 2.8 Antenna pattern magnitude for the weighted delay and sum beamformer, 10 elements spaced 0.1m apart, and no beam steering.

The first null occurs at  $2\pi/(Nd)$ . The repeated main lobes are called *grating lobes*. Figure 2.9 illustrates steering the beam to the  $k_{oy} = \pi/d$  direction.

**2.3.4 Beam Shaping Through Windowing.** The weights in the beamformer perform the function of a multidimensional filter window. In the example above, the constant valued weights correspond to a one dimensional rectangular window, explained pictorially by Fig. 2.10. The window affects the width of the mainlobe and the level of the sidelobes. Ideally, the antenna pattern should be a delta function, i.e. a mainlobe of infinitesimal width and no sidelobes. This translates to the ability to pick out targets no matter how close together they are. However, this is not physically realizable. Using the rectangular window in the previous example produced a reasonably narrow mainlobe resulting in the antenna receiving a small range of directions at the expense of higher

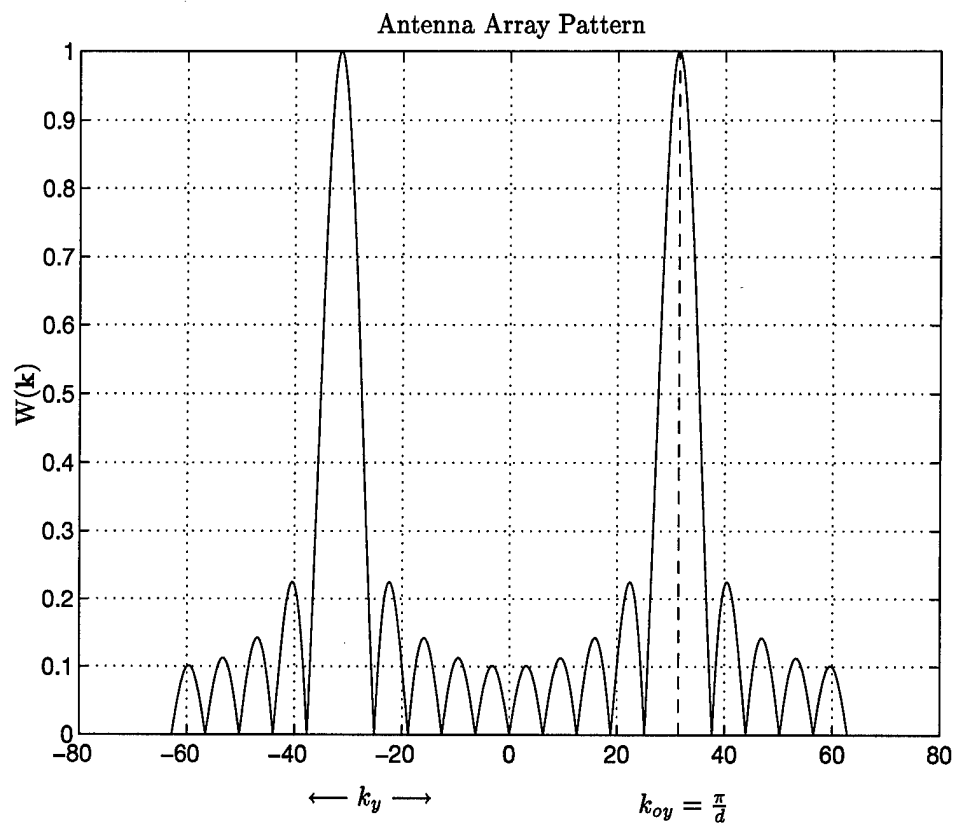


Figure 2.9 Antenna pattern magnitude for the weighted delay and sum beam-former, 10 elements spaced 0.1m apart, with beam steering.

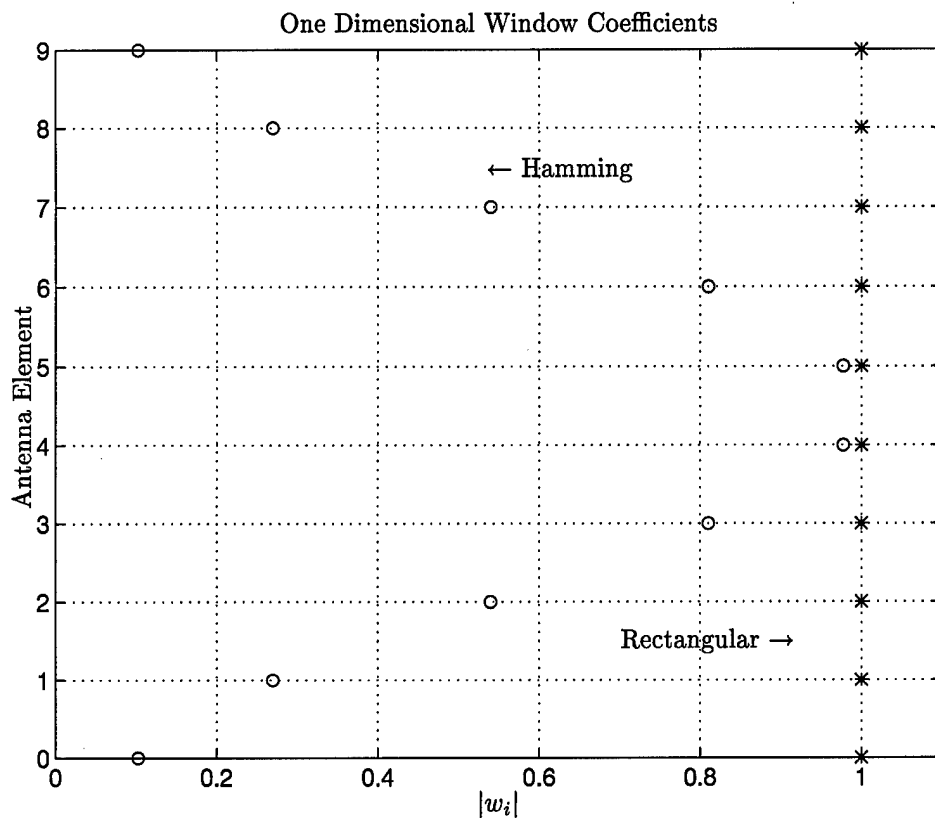


Figure 2.10 Pictorial representation of one dimensional rectangular and Hamming windows used for the receiver weights.

sidelobes (reference Fig. 2.8). The advantage of the rectangular window is simplicity through weights of constant value, shown by the relatively simple expression developed for the antenna pattern in the previous example.

Other windowing options include the Hanning, Hamming, Bartlett, Kaiser, Chebyshev, and Gaussian windows. The expression for the weights using the one dimensional Hamming window [16, page 183] is

$$w_i = 0.54 + (1 - 0.54) \cos \left( \frac{2\pi}{N} (i - N/2) \right). \quad (2.27)$$

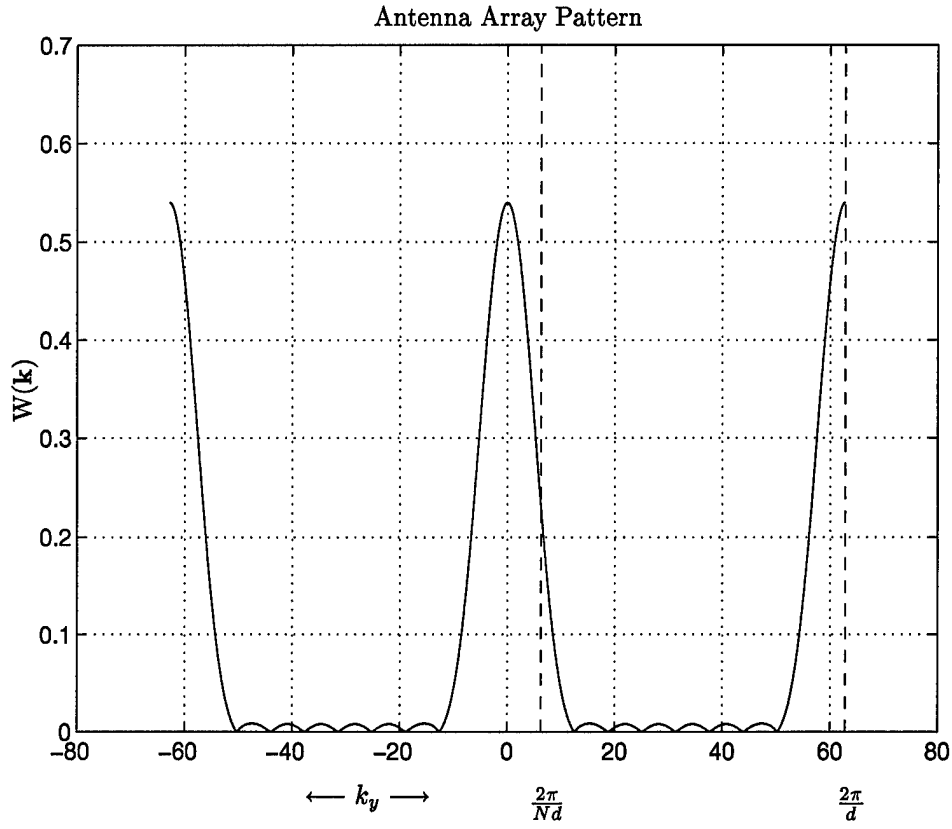


Figure 2.11 Hamming window applied to the weights of a linear antenna array of 10 elements space 0.1m apart.

The shift on the cosine term is added to center the window on the receiver array. The derivation for the previous linear antenna array example using a Hamming window for the weights is

$$\begin{aligned}
 W(\mathbf{k} - \mathbf{k}_o) &= \frac{1}{N} \sum_{i=0}^{N-1} w_i e^{-j(\mathbf{k} - \mathbf{k}_o) \cdot \mathbf{x}_i} \\
 &= \frac{1}{N} \sum_{i=0}^{N-1} \left\{ 0.54 + (1 - 0.54) \cos \left( \frac{2\pi}{N} (i - N/2) \right) \right\} e^{-j(k_y - k_{oy}) \cdot x_{iy}}.
 \end{aligned}$$

Due to the more complicated element weighting, the equation cannot be simplified as in the rectangular window case. Figure 2.11 shows the plot of this expression. Comparing this with Fig. 2.8 highlights the trade-off between a narrow mainlobe and low sidelobes. Also notice the loss in mainbeam gain.

Two dimensional windows can be applied to the weights to control the antenna pattern of a rectangular antenna array. There are also windowing techniques applicable to a hexagonally spaced array, corresponding to a hexagonal sampling lattice in the spatial domain. As long as the antenna array is on a periodic grid, the design of the antenna pattern using multidimensional filters is rather straightforward. However, nonuniform sampling significantly increases the difficulty in designing the receiver weighting function.

#### *2.4 Summary*

In this chapter, the fundamental concepts of beamforming were introduced. The introduction of the plane wave model and antenna beamformer development showed the relevance of signal processing techniques. The wavenumber-frequency plots illustrated the fundamental relationships between the wavenumber-frequency and space-time domains.

From these basic ideas, the weighted delay and sum beamformer was developed and examined. This beamformer serves as an excellent introduction to antenna arrays. Using a linear array example, the antenna pattern was derived and examined. The receiver weights were designed using windowing/signal processing techniques. The use of a different window was shown to reduce the sidelobes of the antenna pattern, at the expense of mainbeam width and gain. It should be noted the weighted delay and sum beamformer can be further generalized by allowing the filter weights to depend on frequency. This configuration is referred to as the filter and sum beamformer.

All of the equations to this point treat time as a continuous variable. Implementation of a beamforming system with digital hardware necessitates sampling the signal temporally. One issue brought forth by the sampling operation is a distortion of the array pattern when the time delays used to steer the beam are not multiples of the sampling period. Dudgeon and Merserau [5, pages 303–307] discuss this in more detail.

### *III. Space-Time Adaptive Processing Primer*

#### *3.1 Introduction*

Airborne radars hold many advantages over ground based radars, however they also suffer several disadvantages. Higher clutter levels due to steep aspect angles to the ground make it difficult to detect and track low flying targets. Airborne radars are also hindered by the constant motion of the airborne platform relative to the ground and the changing terrain, adding a variable nature to the clutter returns. The typical airborne radar environment consists of spectrally variable ground clutter in both the mainlobe and sidelobes of the radar and also interference due to jammers. The aircraft velocity adds Doppler spreading to returns processed by the radar.

Radar engineers have made numerous attempts to improve the detection capabilities of airborne radars. One of the most common is to reduce the effects of clutter through the use of lower sidelobes. However, practical limitations of most antenna designs limit this to the order of only -30 to -40 dB, allowing Doppler spread sidelobe clutter returns to be strong enough to completely mask low-observable targets [12]. Although antennas with lower sidelobe levels are achievable, the costs associated become prohibitive. Also, lower sidelobes do not offer any gains in the ability to detect a target amid jamming.

Another option is non-adaptive filtering. In a ground based radar, reasonably constant clutter patterns allow non-adaptive filtering. However, the Doppler spread and variable nature of the clutter in airborne radars make this option better suited for other applications.

In the last decade, improvements in digital technology make adaptive filtering algorithms very attractive. An adaptive filtering algorithm can have significant impact on radar performance both amid clutter and jamming. Not only is target acquisition improved, but the overall design of the radar is affected. For example, the ability to filter sidelobe clutter reduces the need for extremely low antenna sidelobes. Another example is the Pulse Repetition Frequency (PRF) design trade-off.

Typically the PRF is chosen to help eliminate the effects of clutter, if clutter can be minimized through the use of filtering, the PRF can be chosen to give the radar more unambiguous range.

Space-Time Adaptive Processing (STAP) refers to this class of adaptive filtering algorithms. Using digital beamforming concepts, STAP is defined by Ward [21] as the simultaneous combination of signals from the elements of an array antenna (the spatial domain) and the multiple pulses of a coherent radar waveform (the temporal domain) to suppress interference and provide target detection. STAP is inherently an extension of digital beamforming architecture and principles.

Chapter II introduced the basics of beamforming and serves as the foundation for the principles encountered in this next development. As explained, the beamforming operation occurred through the use of weights on each antenna element. The output of the summed and weighted elements resulted in a single data stream containing signals from propagating waves in the direction of interest. STAP differs from basic antenna beamforming in that no weights are used on the elements themselves. The received space-time signal from each element is stored for later processing. The weighting functions are applied as part of the processing algorithm.

The development begins by defining the radar geometry. Models are introduced for noise, jamming, and clutter along with the target return signal. The nomenclature and architecture of STAP is introduced as the development progresses. Chapter II concepts are tied together to form these models. Ward's [21] work is the basis for the developments in this chapter.

### *3.2 The Physical and Electromagnetic Environment*

*3.2.1 Physical Geometry.* The radar antenna array for all examples in this chapter is assumed linear with uniformly spaced elements a distance  $d$  apart, each with their own receiver. The orientation of the array is along the  $x$ -axis, as shown in Fig. 3.1a. Therefore, the vector to the



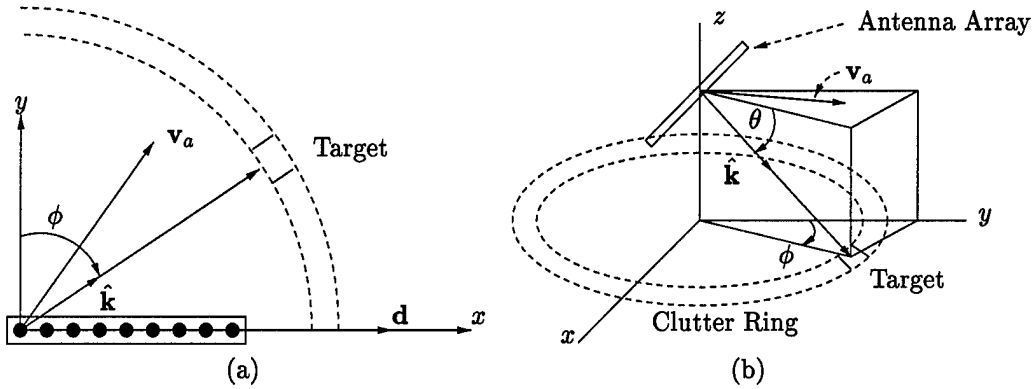


Figure 3.1 (a) Uniformly spaced antenna element array geometry and (b) radar platform geometry.

first antenna element is

$$\mathbf{d} = d\hat{\mathbf{x}}, \quad (3.1)$$

where  $\hat{\mathbf{x}}$  is the unit vector in the  $x$  axis direction. Using this expression, the vector to the  $n^{th}$  element is

$$\mathbf{r}_n = n\mathbf{d}. \quad (3.2)$$

As used in Fig. 3.1b, the vector  $\mathbf{v}_a$  shows the magnitude and direction of the platform velocity. This figure also shows the corresponding radar platform geometry.

The unit vector  $\hat{\mathbf{k}}(\phi, \theta)$  shown in Fig. 3.1b represents the direction to a point of interest (or target). This direction corresponds to the direction of  $\mathbf{k}$ , the spatial frequency vector from Chapter II, see Eqn. (2.1). The angular variables  $\phi$  and  $\theta$  respectively represent the azimuth and elevation to the point of interest (target, clutter, jammer, etc.). The relationship of Ward's [21] coordinate system to the Cartesian coordinate system is described by

$$\hat{\mathbf{k}}(\phi, \theta) = \cos \theta \sin \phi \hat{\mathbf{x}} + \cos \theta \cos \phi \hat{\mathbf{y}} + \sin \theta \hat{\mathbf{z}}, \quad (3.3)$$

where  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{z}}$  again represent unit vectors along the Cartesian coordinate system  $\hat{\mathbf{y}}$  and  $\hat{\mathbf{z}}$  axes.

**3.2.2 Signal and Radar Description.** Now that the physical geometry of the radar is defined, let us examine the electromagnetic properties of the radar. The radar is assumed to be Pulse Doppler (PD), therefore target velocity information is available to the processor.

The sinusoidal waveform transmitted by the radar is modeled mathematically by

$$\tilde{s}(t) = a_t u(t) e^{j(\omega_o t + \psi)}, \quad (3.4)$$

where the complex exponential and random phase shift  $\psi$  represent the sinusoidal carrier,  $a_t$  the transmitted pulse amplitude, and  $u(t)$  the envelope function. The envelope function defines the pulse width and Pulse Repetition Interval (PRI) characteristics. A series of individual pulses summed to form the pulse train of the radar defines the envelope function,

$$u(t) = \sum_{m=0}^{M-1} u_p(t - mT_r). \quad (3.5)$$

The fundamental pulse is time shifted by multiples of the PRI,  $T_r$ . The summation of a finite number of terms,  $M$ , shows the *Coherent Processing Interval*, or CPI. This expression implicitly defines  $M$  as the number of pulses in the CPI.

One should be careful to distinguish between the coherent processing interval and the coherent integration time. We will find later that the output of the STAP algorithm is described by  $y = \mathbf{w}^H \chi$ , where  $\mathbf{w}$  is the adaptive weight vector and  $\chi$  is the incoming space-time snapshot (discussed later in detail). This expression indicates the coherent integration time since the inner product of the weight vector and space-time snapshot is simply the weighted sum of the incoming data samples, a form of integration.

The CPI defines how much data the adaptive processor uses in determining the adapted pattern. Obviously, a smaller CPI allows rapid modifications to the antenna pattern in response to the changing environment. However, making it too small does not give the processor enough information about the statistics of the environment, producing poorly formed antenna patterns. Making the CPI too large results in slow response. This is one of the primary design considerations in a STAP system.

The transmit signal amplitude  $a_t$  is defined to capture the energy in the pulse. Therefore, the fundamental pulse energy  $E_u$  is normalized to one,

$$E_u = \int_0^{T_r} |u_p(t)|^2 dt = 1. \quad (3.6)$$

Consequently, the total energy transmitted in a single pulse  $E_p$  meets the definition for  $a_t$ ,

$$E_p = \int_0^{T_r} |\tilde{s}(t)|^2 dt = a_t^2. \quad (3.7)$$

From this expression, the energy within one CPI is easily determined by integrating over the CPI length. Since the focus of STAP algorithms is the response and pattern after one CPI, it is customary to consider only the signal transmitted within one CPI as being the entire transmitted signal. Hence, the energy within one CPI is synonymous with the transmitted signal energy  $E_t$ ,

$$E_t = \int_0^{MT_r} |\tilde{s}(t)|^2 dt = M a_t^2 = M E_p. \quad (3.8)$$

The signal received by each individual antenna element is the transmitted signal (Eqn. (3.4)) with a time delay from the target to the  $n^{th}$  element,  $\tau_n$ , and a Doppler shift  $f_t$  due to the target's

relative velocity  $v_t$  between the target and the platform,  $f_t = 2v_t/\lambda_o$ ,<sup>1</sup>

$$\tilde{s}_n(t) = \tilde{s}(t - \tau_n)e^{j2\pi f_t(t - \tau_n)}. \quad (3.9)$$

Inserting the transmitted pulse expression gives

$$\tilde{s}_n(t) = a_r u(t - \tau_n)e^{j2\pi f_o(t - \tau_n)}e^{j2\pi f_t(t - \tau_n)}e^{j\psi}, \quad (3.10)$$

where  $a_r$  represents the received pulse amplitude. This parameter is introduced because the atmosphere and Radar Cross Section (RCS) of the target attenuate the amplitude of the pulse, hence it is typically much less than  $a_t$ .

If the Doppler shift is greater than the Pulse Repetition Frequency (PRF),  $f_t > f_r$ , then the velocity of the target is ambiguous. This is shown in Fig. 3.2. Once the target Doppler shift has increased past the first PRF line located at  $f_r$ , the radar can no longer unambiguously determine the target velocity. For this reason, target Doppler frequency is normalized to the PRF,

$$\bar{\omega}_t = \frac{f_t}{f_r} = f_t T_r. \quad (3.11)$$

Using the normalized Doppler frequency  $\bar{\omega}$ , a quick glance determines if the Doppler shift is unambiguous (normalized Doppler less than one) or ambiguous (normalized Doppler greater than one).

Similar to Chapter II, the signal delay is expressed as a delay from the antenna element of interest to the reference element (the first element to receive the return signal),  $\tau'_n$ , and a delay

---

<sup>1</sup>The Doppler shift is assumed equal at all antenna elements.

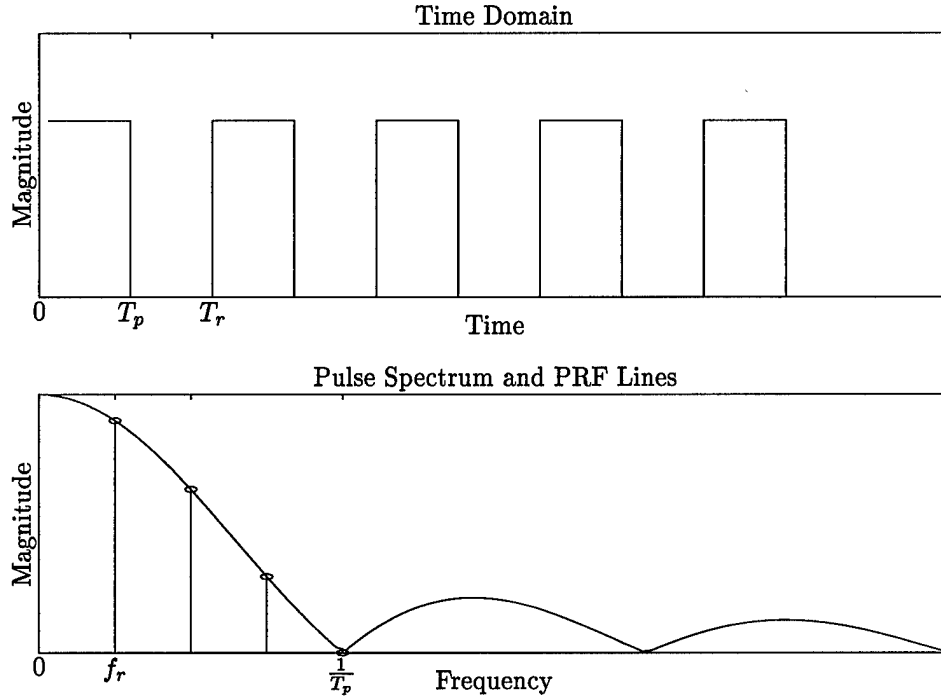


Figure 3.2 Time domain and frequency domain plots of the transmitted pulse illustrating ambiguous Doppler frequency.

equal to the round trip time to the target,  $\tau_t$ ,

$$\tau_n = \tau_t + 2\tau'_n. \quad (3.12)$$

Keeping the physical geometry of the scenario under consideration, we see the  $n^{th}$  antenna element's delay is a function of the target's elevation  $\theta_t$ , azimuth  $\phi_t$ , and range  $R_t$ . The round trip time to the target is a simple function of the velocity of propagation  $c$  and range  $R_t$ ,

$$\tau_t = \frac{2R_t}{c}. \quad (3.13)$$

The differential delay from the reference element to the  $n^{th}$  element is

$$\tau'_n = -\frac{\hat{\mathbf{k}}(\theta_t, \phi_t) \cdot \mathbf{r}_n}{c} = -\frac{n\hat{\mathbf{k}}(\theta_t, \phi_t) \cdot \mathbf{d}}{c} = -n\frac{d}{c} \cos \theta_t \sin \phi_t. \quad (3.14)$$

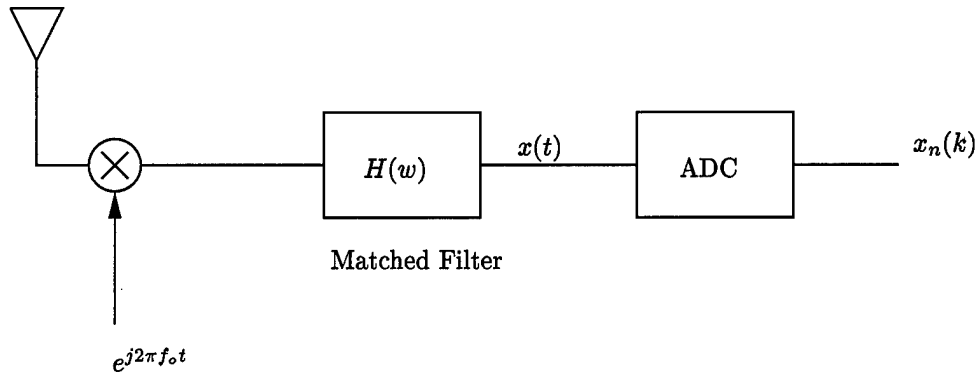


Figure 3.3 Antenna array element block diagram. ADC is an analog to digital converter.

Related to this expression is the target spatial frequency,

$$\vartheta_t = \frac{\hat{\mathbf{k}}(\theta_t, \phi_t) \cdot \mathbf{d}}{\lambda_o} = \frac{d}{\lambda_o} \cos \theta_t \sin \phi_t. \quad (3.15)$$

The above equation for the spatial frequency relates directly to the  $|\mathbf{k}_o|$  found in Chapter II Eqn. (2.3) for an array oriented along the  $x$  axis.

Using these expressions, the received signal shown in Eqn. (3.10) can be rearranged to show the radar *ambiguity function*. Since each antenna element has its own receiver and matched filter as shown in Fig. 3.3, the matched filter output<sup>2</sup> for the  $n^{\text{th}}$  element signal is given as [21],

$$x_n(t) = a_r e^{j\psi} e^{jn2\pi\vartheta_t} \sum_{m=0}^{M-1} e^{jm2\pi\bar{\omega}_t} \chi(t - \tau_t - mT_r, \bar{\omega}_t), \quad (3.16)$$

where  $\chi(\tau_t, f_t)$  is the radar ambiguity function defined by Skolnik [17, page 412] as

$$|\chi(\tau_t, f_t)|^2 = \left| \int_{-\infty}^{\infty} u(\beta) u^*(\beta + \tau_t) e^{j2\pi f_t \beta} d\beta \right|^2. \quad (3.17)$$

The ambiguity function is used to assess the properties of the transmitted waveform in terms of its target resolution, measurement accuracy, ambiguity, and response to clutter [17]. Section 11.4

<sup>2</sup>Ward's expression assumes  $\tau_t \gg 2\tau'_n$ .

of Skolnik's text [17] is an excellent introduction to the usefulness of the radar ambiguity function. For the purposes of this discussion, the following property is of interest.

The ambiguity function evaluated at (0,0) represents twice the energy in the return signal. Because of the waveform normalization shown in Eqn. (3.6) this is simply one,

$$|\chi(0,0)|^2 = (2E)^2 = 1. \quad (3.18)$$

Let us consider the return from a target at a particular range gate,  $l$ . First, discretize time through a sampling operation at each PRI. In other words, the continuous variable  $t$  becomes

$$t_m = \tau_t + mT_r \text{ for } m = 0 : M - 1, \quad (3.19)$$

where the round trip delay  $\tau_t$  is introduced to simplify the next step. The sampled target return from a particular range gate is

$$x_n(t_m) = x_{nm} = a_r e^{j\psi} e^{j2\pi n \vartheta_t} \sum_{q=0}^{M-1} e^{j2\pi q \bar{\omega}_t} \chi((m-q)T_r, f_t), \quad (3.20)$$

where  $\bar{\omega}_t$  is the normalized Doppler frequency and the range gate dependence,  $l$ , has been suppressed from the subscript. From this point forward, we always assume the item of interest is for a particular range gate/cell allowing suppression of the  $l$  subscript. It is assumed no target range ambiguities exist, therefore the  $\chi$  function is zero everywhere except when  $q = m$ . This allows the simplification

$$x_{nm} = a_r e^{j\psi} e^{j2\pi n \vartheta_t} e^{j2\pi m \bar{\omega}_t} \chi(0, f_t). \quad (3.21)$$

Furthermore, the product of the target Doppler shift and pulse width is assumed much less than one allowing the approximation

$$f_t T_p \ll 1 \text{ implies } \chi(0, f) \approx 1. \quad (3.22)$$

Using this approximation, the target samples at the  $n^{th}$  antenna element and  $m^{th}$  pulse,  $x_{nm}$  can be expressed as

$$x_{nm} = \alpha_t e^{j2\pi n \vartheta_t} e^{j2\pi m \bar{\omega}_t}, \quad (3.23)$$

where the complex random amplitude term  $\alpha_t$  contains the phase and amplitude terms shown previously, i.e.  $\alpha_t = a_r e^{j\psi}$ .

### 3.3 Fundamentals of STAP

As mentioned previously in Chapter II, MATLAB<sup>®</sup> notation is used wherever possible to express the formulation of matrices and vectors. In particular,  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2]$  places the two column vectors side by side to form a matrix, and  $\chi = \text{vec}(\mathbf{X})$  places the columns of  $\mathbf{X}$  end-to-end forming a vector. Lower case bold letters represent vectors and upper case bold represent matrices.

To begin the fundamentals of STAP, examine Eqn. (3.23). First, create a column vector containing each antenna element sample at a particular range gate  $l$  and pulse  $m$  denoting it  $\mathbf{x}_m$ ,

$$\begin{aligned} \mathbf{x}_m &= \alpha_t e^{j2\pi m \bar{\omega}_t} [1; e^{j2\pi \vartheta_t}; e^{j4\pi \vartheta_t}; \dots; e^{j2(N-1)\pi \vartheta_t}]^T \\ &= \alpha_t e^{j2\pi m \bar{\omega}_t} \mathbf{a}_t(\vartheta_t), \end{aligned} \quad (3.24)$$

where  $\mathbf{a}_t(\vartheta_t)$  is the *spatial steering vector* of length  $N$  and the range gate/cell dependence has again been suppressed in the subscript. Using this notation, the incoming samples can be written as the



matrix  $\mathbf{X}$ ,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_{M-1} \end{bmatrix}, \quad (3.25)$$

where the matrix is for the range gate/cell of interest.

Since the spatial frequency is a function of the target elevation and azimuth angles (see Eqn. (3.15)), the spatial steering vector can be written as an explicit function of these variables,

$$\mathbf{a}_t(\theta_t, \phi_t) = [1 \ e^{j\frac{2\pi d}{\lambda_o} \cos \theta_t \sin \phi_t} \ e^{j\frac{4\pi d}{\lambda_o} \cos \theta_t \sin \phi_t} \ \dots \ e^{j\frac{2(N-1)\pi d}{\lambda_o} \cos \theta_t \sin \phi_t}]^T. \quad (3.26)$$

Similarly, the *temporal steering vector*  $\mathbf{b}_t(\bar{\omega}_t)$  is defined as the vector of exponentials spanning the  $M$  pulses in the CPI,

$$\mathbf{b}_t(\bar{\omega}_t) = [1 \ e^{j2\pi\bar{\omega}_t} \ e^{j4\pi\bar{\omega}_t} \ \dots \ e^{j2(M-1)\pi\bar{\omega}_t}]^T. \quad (3.27)$$

The relationship between these two vectors forming the matrix of received target samples is characterized by the *Kronecker* product, denoted  $\otimes$ . See Appendix A for a discussion of its properties. The Kronecker product of the temporal and spatial steering vectors,

$$\mathbf{v}_t(\bar{\omega}_t, \vartheta_t) = \mathbf{b}_t(\bar{\omega}_t) \otimes \mathbf{a}_t(\vartheta_t), \quad (3.28)$$

is defined as the *space-time steering vector*  $\mathbf{v}_t$ , where the above equation emphasizes it's normalized Doppler and spatial frequency dependence. Scaling this length  $NM$  vector by a random received amplitude parameter (subscripted with a  $t$  for the target) gives

$$\chi_t = \alpha_t \mathbf{v}_t, \quad (3.29)$$

where  $\chi_t$  is the *space-time snapshot* for the target. This length  $NM$  column vector represents the target returns received by each antenna element ( $n = 0 : N - 1$ ) due to each transmitted pulse ( $m = 0 : M - 1$ ) at a particular range gate/cell of interest  $l$ . Relating this Kronecker product to the matrix of returns shown in Eqn. (3.25) illustrates it is simply the stacking of the columns of  $\mathbf{X}$  end to end to form the column vector,

$$\chi = \text{vec}(\mathbf{X}). \quad (3.30)$$

Since the target is not the only echo returned to the radar, an *undesired* component,  $\chi_u$  must be introduced in the model,

$$\chi = \chi_t + \chi_u. \quad (3.31)$$

Our model will consider the undesired components due to clutter, noise, and barrage noise jamming.

Let us give a quick summary of the important points of the development so far. The complex scalar  $x_{nm}$  contains the received signal from a particular antenna element and pulse of interest. Expanding this into a vector across all of the antenna elements results in the spatial steering vector  $\mathbf{a}(\vartheta)$  containing the received signals from every antenna element at the pulse of interest. Expanding  $x_{nm}$  into a vector across all pulses results in the temporal steering vector  $\mathbf{b}(\bar{\omega})$  containing the received signals from every pulse but only one antenna element. The space-time snapshot combines these two vectors into one column vector, denoted by the use of the Kronecker Product notation. It contains all of the received signals from every pulse and every antenna element at one range gate  $l$ . Since the time between each pulse is divided into  $L$  range bins, there are  $L$  total space-time snapshots in the CPI.

Figure 3.4 is a visual illustration of the CPI datacube. The shaded cross section is representative of the space-time snapshot where the columns of the matrix/cross section form the snapshot

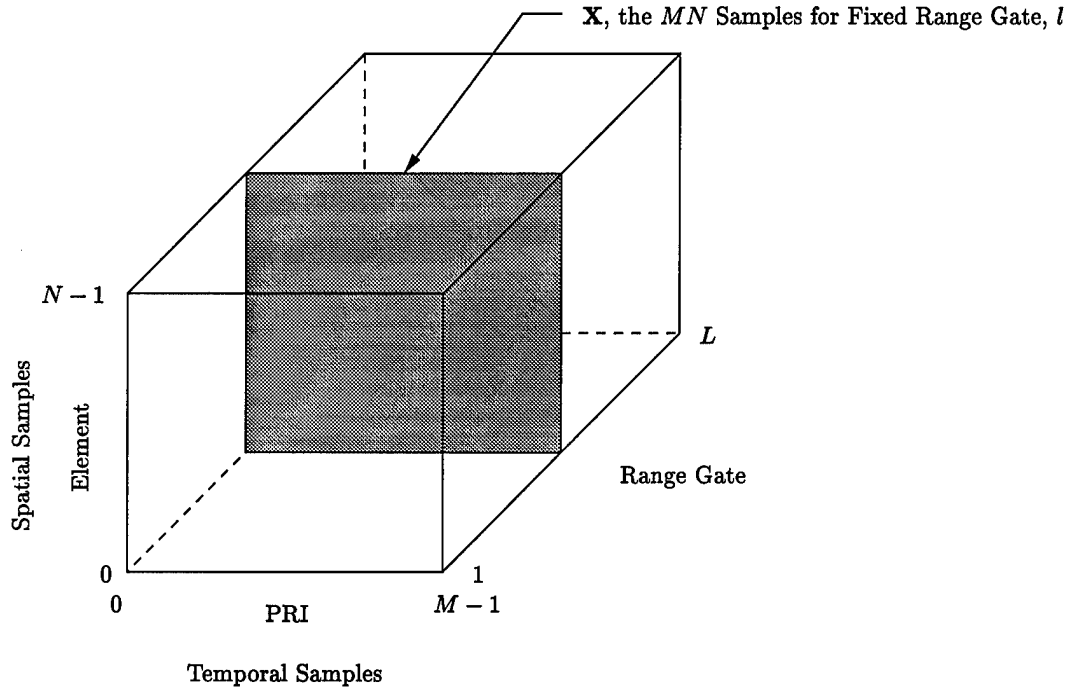


Figure 3.4 The CPI datacube of dimension  $M \times N \times L$ .

vector. As mentioned previously, the number of pulses  $M$  in the CPI define its length. Therefore, the time duration of the CPI is  $MT_r$ .

Also, notice the raw data is stored from each antenna element. No weighting is applied to the received signals as explained in Chapter II. The weighting is performed after the data is stored. This is the function of the space-time adaptive processor. Withholding the weights until after the data in the CPI is known allows the statistics of the data to drive their calculation.

The next step in the development of STAP fundamentals is analysis of the undesired component  $\chi_u$  in Eqn. (3.31). This component consists primarily of three sources: thermal noise, jammers, and clutter. The following sections examine each of these in detail.

**3.3.1 Thermal Noise Model.** This model begins by assuming the only noise present is due to the internally generated thermal noise of the receiver. This noise is taken to be white, therefore there is no correlation of the noise between pulses and it is also uncorrelated spatially (between antenna elements). Let  $x_{nm}$  describe the noise received in the  $n^{th}$  antenna element. Since the noise

is assumed white, the expected value of received noise is

$$\mathcal{E}\{x_{nm}x_{n'm}^*\} = \sigma^2\delta_{n-n'}, \quad (3.32)$$

where  $\sigma^2$  represents the noise power in a single element. This noise power is given by the noise power spectral density  $N_o$  multiplied by the bandwidth  $B$  of the receiver (since this analysis assumes a separate receiver on each antenna element),  $\sigma^2 = N_o B$ . The Kronecker delta  $\delta$  in the above equation is defined as

$$\delta_{n-n'} \triangleq \begin{cases} 1 & n = n' \\ 0 & n \neq n'. \end{cases} \quad (3.33)$$

A similar expression shows the correlation of the noise terms from pulse to pulse,

$$\mathcal{E}\{x_{nm}x_{nm'}^*\} = \sigma^2\delta_{m-m'}. \quad (3.34)$$

If we let  $\chi_n$  represent the  $MN \times 1$  spatial snapshot of noise signals on each antenna element, the noise space-time covariance matrix  $\mathbf{R}_n$  is shown by

$$\mathbf{R}_n = \mathcal{E}\{\chi_n\chi_n^H\} = \sigma^2\mathbf{I}_{MN} = \sigma^2\mathbf{I}_M \otimes \mathbf{I}_N, \quad (3.35)$$

where  $\mathbf{I}$  is the identity matrix with dimension shown by the subscript and  $\sigma^2$  is the previously defined noise power per antenna element. The covariance matrix shown above serves as the thermal noise component of the overall covariance matrix  $\mathbf{R}$ . The next undesired component considered is the jamming model.

**3.3.2 Jamming Model.** At this point, only noise jamming originating from a long range airborne or ground based platform is considered. The bandwidth of the jammer  $B_j$  is assumed

larger than the radars instantaneous bandwidth (the inverse of the time domain pulse width  $T_p$ ), i.e.

$$B_j > B = \frac{1}{T_p}. \quad (3.36)$$

In order to counter a jammer through STAP, its signal must remain correlated across the antenna elements. This means the time for the jamming signal to travel across the array is less than the inverse of its bandwidth, a measure of the correlation time for the jammer. Figure 3.5 helps explain this point.

Examining the inverse Fourier transform of the jammer's power spectral density envelope shows an increase in the time domain width as the jammer bandwidth *decreases*. Conversely, as the bandwidth of the jammer approaches infinity the time domain width approaches a delta function. The null-to-null width of the time domain  $\text{sa}(x)$  (sine over argument) curve can be viewed as the time in which the statistics of the jammer remain correlated. Therefore, if the time to transcend the antenna array is less than this width the jammer signal remains correlated. In other words, the relationship

$$\frac{1}{B_j} > \frac{Nd}{c} \quad (3.37)$$

must hold for the jammer to remain correlated across the array.

The jammer signal received by each antenna element is analyzed in the same manner as the target signal. Again, the ultimate goal is development of a space-time covariance matrix  $\mathbf{R}_j$  for use in the STAP algorithm. This covariance matrix contains the correlation statistics of the jammer and allows the algorithm to locate the jammer spatially and in time. A null is then formed in that location.

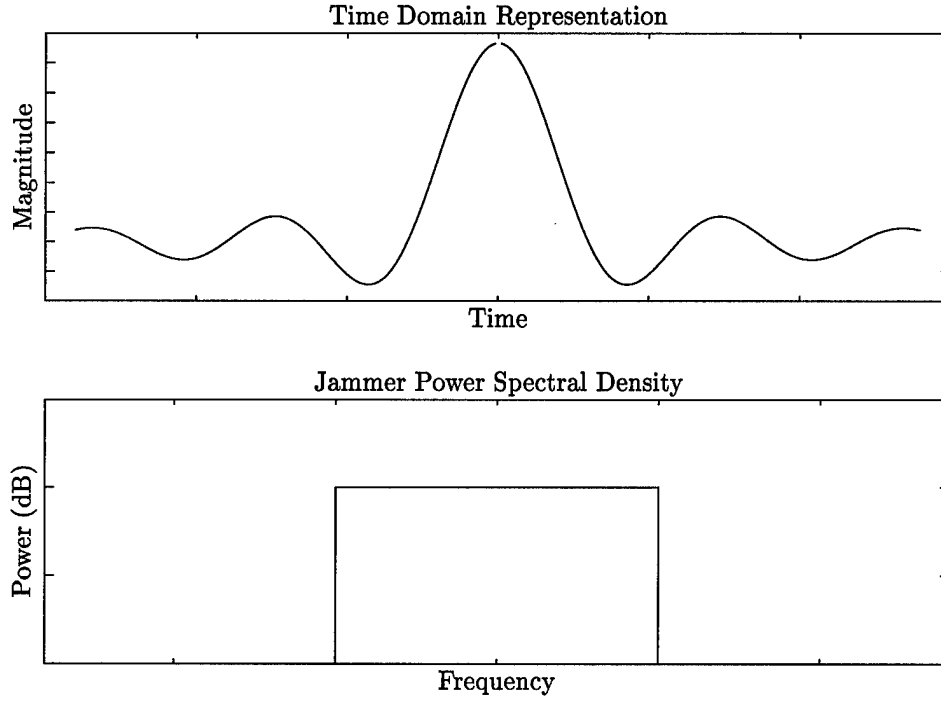


Figure 3.5 The jammer bandwidth time domain correlation relationship.

First, let  $x_{nm}$  represent the signal received from the jammer on the  $n^{th}$  antenna element at the  $m^{th}$  pulse. Creating a vector of samples across the antenna array elements results in

$$\mathbf{x}_m = \alpha_m \mathbf{a}_j(\theta_j, \phi_j), \quad (3.38)$$

where the jammer amplitude is represented by  $\alpha_m$  and  $\mathbf{a}_j$  is the jammer spatial steering vector. Since this is a noise jammer the amplitude is random. This amplitude fluctuates from pulse-to-pulse and can be represented as a vector of random amplitudes,

$$\boldsymbol{\alpha}_j = [\alpha_0 \ \alpha_1 \ \dots \ \alpha_{M-1}]^T. \quad (3.39)$$

As before, the Kronecker Product allows the compact notation of a space-time snapshot for the jammer  $\chi_j$ ,

$$\chi_j = \alpha_j \otimes \mathbf{a}_j. \quad (3.40)$$

The jammer is assumed correlated across the antenna array as mentioned in the beginning of this section. Furthermore, the jammer location is assumed constant throughout a CPI. Using these assumptions, the random amplitudes are uncorrelated as shown by

$$\mathcal{E}\{\alpha_m \alpha_{m'}^*\} = \sigma^2 \xi_j \delta_{m-m'}, \quad (3.41)$$

where  $\delta$  is again the Kronecker delta and  $\xi_j$  is the Jammer-to-Noise Ratio (JNR). Writing the above equation in vector form results in

$$\mathcal{E}\{\alpha \alpha^H\} = \sigma^2 \xi_j \mathbf{I}_M, \quad (3.42)$$

where the parameter  $\sigma^2$  is noise power per element as mentioned in the noise model section.

The JNR is easily derived using the radar range equation as given by Skolnik [17]. First examine the one way range equation for the received jammer power

$$J_o = \frac{S_j A_e}{4\pi R_j^2} \quad (3.43)$$

given  $S_j$  (W/Hz) is the radiated power spectral density (hence it already includes the effect of the jammer antenna gain),  $A_e$  is the effective area of the receive antenna, and  $R_j$  is the range to the jammer. The effective area of the receive antenna depends on gain and wavelength as given by

$$A_e = \frac{\lambda_o^2 G_r}{4\pi}, \quad (3.44)$$

where  $G_r$  is the gain of the receive antenna and  $\lambda_o$  is the nominal wavelength. Substituting for the receive antenna effective area in Eqn. (3.43) gives

$$J_o = \frac{S_j G_r \lambda_o^2}{(4\pi)^2 R_j^2}. \quad (3.45)$$

The receive antenna elements are assumed to radiate with a power pattern given by  $g(\theta, \phi)$ , or the square of the voltage pattern  $|f(\theta, \phi)|^2$ . The term  $L_r$  is introduced to account for losses in the receiver and is essentially a scaling factor. Using these tools, the received jammer power spectral density  $J_o$  becomes

$$J_o = \frac{S_j g(\theta, \phi) \lambda_o^2}{(4\pi)^2 R_j^2 L_r} \quad (3.46)$$

and the space-time covariance matrix for the jammer  $\mathbf{R}_j$  results through the following manipulation,

$$\begin{aligned} \mathbf{R}_j &= \mathcal{E} \{ \chi_j \chi_j^H \} \\ &= \mathcal{E} \{ (\alpha_j \otimes \mathbf{a}_j)(\alpha_j \otimes \mathbf{a}_j)^H \} \\ &= \mathcal{E} \{ \alpha_j \alpha_j^H \otimes \mathbf{a}_j \mathbf{a}_j^H \} \\ &= \sigma^2 \xi_j \mathbf{I}_M \otimes \mathbf{a}_j \mathbf{a}_j^H. \end{aligned} \quad (3.47)$$

Again, this relationship assumes the jammer location is constant over the CPI.

Let us extend Eqn. (3.47) to two jammer signals. The vector of received jamming signals becomes

$$\mathbf{x}_m = \alpha_m^{(1)} \mathbf{a}_j \left( \theta_j^{(1)}, \phi_j^{(1)} \right) + \alpha_m^{(2)} \mathbf{a}_j \left( \theta_j^{(2)}, \phi_j^{(2)} \right), \quad (3.48)$$



where the superscripts denote the jammer. The jammers are assumed independent, therefore the cross correlation of the amplitudes is zero

$$\mathcal{E} \left\{ \alpha_m^{(l)} \alpha_{m'}^{(l')} \right\} = \sigma^2 \xi_j^{(l)} \delta_{l-l'} \delta_{m-m'}. \quad (3.49)$$

Forming the space-time snapshot is merely a matter of summing the signals received from the two jammers,

$$\chi_j = \alpha_j^{(1)} \otimes \mathbf{a}_j^{(1)} + \alpha_j^{(2)} \otimes \mathbf{a}_j^{(2)}. \quad (3.50)$$

The space-time covariance matrix then follows directly,

$$\begin{aligned} \mathbf{R}_j &= \mathcal{E} \{ \chi_j \chi_j^H \} \\ &= \mathbf{R}_j^{(1)} + \mathbf{R}_j^{(2)} \\ &= \mathbf{I}_m \otimes \sigma^2 \xi_j \left( \mathbf{a}_j^{(1)} \mathbf{a}_j^{(1)H} + \mathbf{a}_j^{(2)} \mathbf{a}_j^{(2)H} \right) \\ &= \mathbf{I}_m \otimes \left( \Phi_j^{(1)} + \Phi_j^{(2)} \right), \end{aligned} \quad (3.51)$$

where the two  $N \times N$  matrices  $\Phi_j^{(1)}$  and  $\Phi_j^{(2)}$  represent the jammer spatial covariance matrices. They are the result of taking the expected value of the received jammer signal across the elements at a particular pulse,

$$\Phi_j = \mathcal{E} \{ \mathbf{x}_m \mathbf{x}_m^H \} = \sigma^2 \xi_j \mathbf{a}_j \mathbf{a}_j^H. \quad (3.52)$$

Now take a look at the more general case of  $P$  jammers. Since the result is again a sum of the individual jammer space-time covariance matrices, Eqn. (3.51) still holds except there are now

$P$  terms. However, it can be more compactly represented by redefining  $\Phi$  to represent all of them,

$$\Phi = \mathbf{A}_j \Xi_j \mathbf{A}_j^H, \quad (3.53)$$

where  $\mathbf{A}_j$  is an  $N \times N$  matrix whose columns contain each of the jammers' spatial steering vectors as given by

$$\mathbf{A}_j = [\mathbf{a}_j^{(1)} \ \mathbf{a}_j^{(2)} \ \dots \ \mathbf{a}_j^{(P)}] \quad (3.54)$$

and  $\Xi_j$  is a  $P \times P$  matrix containing the received signal-to-noise ratio for each jammer multiplied by the element noise power along its diagonal, defined by

$$\Xi_j = \sigma^2 \text{diag}(\xi_j). \quad (3.55)$$

Therefore, the jammer space-time covariance matrix for  $P$  jammers becomes

$$\mathbf{R}_j = \mathbf{I}_M \otimes \Phi. \quad (3.56)$$

**3.3.2.1 Jammer Space-Time Covariance Matrix Rank.** The rank of the jammer space-time covariance matrix is important in the design of STAP techniques. If the matrix is of low rank the STAP algorithm expends fewer degrees of freedom placing nulls on the jammers.

The rank of the space-time covariance matrix for a single jammer is equal to  $M$  since the outer product shown in Eqn. (3.47) is a linear combination of the vectors. However, for  $P$  jammers the rank increases considerably.

Since the size of  $\mathbf{R}_j$  is  $MN \times MN$  regardless of the number of jammers, the worst case scenario is a matrix of full rank equal to  $(MN)^2$ . However, because  $\mathbf{R}_j$  is block diagonal (see Eqn. (3.56)),

the rank of  $\mathbf{R}_j$  is a linear combination of the rank of each of the blocks along the diagonal or

$$\text{rank}(\mathbf{R}_j) = M \text{ rank}(\Phi). \quad (3.57)$$

The problem of determining the jammer rank now relies on the rank of the jammer spatial covariance matrix  $\Phi$ . The following relationship between the dimension of two matrices holds [19, page 200],

$$\dim(\mathbf{A} + \mathbf{B}) = \dim(\mathbf{A}) + \dim(\mathbf{B}) - \dim(\mathbf{A} \cap \mathbf{B}). \quad (3.58)$$

The spatial covariance matrix  $\Phi$  representing  $P$  jammers is a sum of the individual spatial covariance matrices (each of rank one). Given statistically independent jammers (different locations), the dimension of the intersection of the two matrices is zero. Therefore the dimension (or rank) of the jammer spatial covariance matrix  $\Phi$  becomes

$$\text{rank}(\Phi) = P, \quad (3.59)$$

where  $P$  is the number of jammers. Using this fact, the rank of the jammer space-time covariance matrix is seen to be significantly lower than full rank,

$$\text{rank}(\mathbf{R}_j) = MP. \quad (3.60)$$

Therefore  $MP$  Degrees Of Freedom (DOF) are used to place nulls on the jammers where  $P < N$ .

**3.3.3 Clutter Model.** Clutter is defined as any unwanted radar echo. Because the focus of this section is on airborne surveillance radars, the Earth serves as the primary source of clutter. Hence, the model described in this section only describes clutter due to the surface of the Earth.

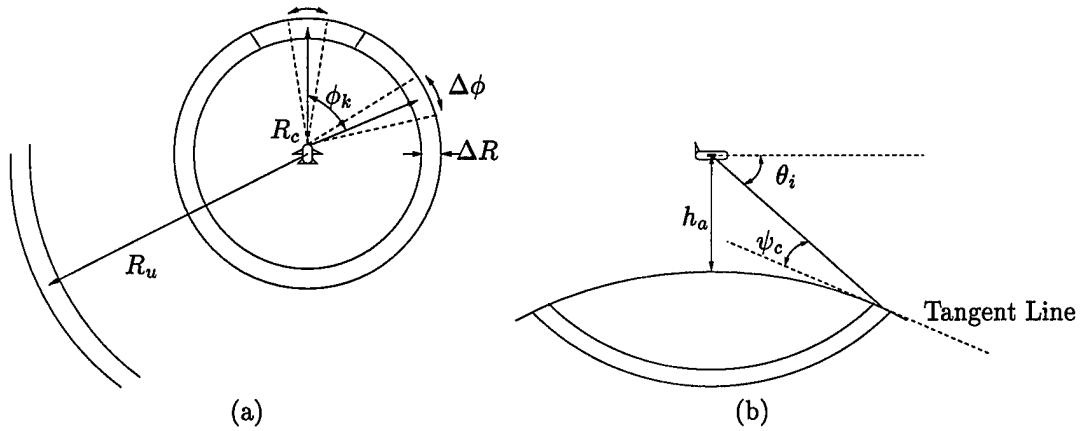


Figure 3.6 Top and side views of the clutter ring.

The return from clutter is very similar to that of a target. However, several key differences play an important role in the derivation of the model. First, since the clutter is due to the surface of the Earth it is distributed in range to the horizon. Second, the surface of the Earth is considered stationary. Hence, the only Doppler shift is due to the velocity of the airborne platform relative to the clutter patch of interest. Using these observations, the clutter is broken into rings of constant range around the aircraft. Further separation of the clutter ring results in clutter patches. Each clutter patch is considered of constant range and velocity with respect to the airborne platform. Figure 3.1b shows the clutter ring for the airborne platform. A top view of the clutter ring (with the airborne platform at the center) is shown in Fig. 3.6a.

The unambiguous range is divided into  $L$  range bins (rings of constant range) and each ring into  $N_c$  patches. The elevation angle to the  $i^{th}$  clutter ring/range bin is given by

$$\theta_i = -\sin^{-1} \left( \frac{R_c^2 + h_a(h_a + 2a_e)}{2R_c(a_e + h_a)} \right), \quad (3.61)$$

where  $R_c$  denotes the range to the clutter ring,  $h_a$  the altitude, and  $a_e$  the effective radius of the Earth. As is customary, the effective radius is considered as  $\frac{4}{3}$  the mean radius ( $\frac{4}{3}a$ ) of the Earth to account for the refraction of radio waves in the atmosphere [17, page 449]. As defined in Fig. 3.1b, the elevation angle is referenced to the horizontal from the airborne platform.

The grazing angle  $\psi_c$  is the angle between a line connecting the airborne platform to the horizon and a line tangent to the earth at the horizon point. Shown in Fig. 3.6b, this angle is given by

$$\psi_c = -\sin^{-1} \left( \frac{R_c^2 - h_a(h_a + 2a_e)}{2R_c a_e} \right). \quad (3.62)$$

The grazing angle is zero when the clutter ring of interest is equal to the range to the radar horizon.

The spatial frequency of the  $ik^{th}$  clutter patch<sup>3</sup> is found by substituting the angles associated with the patch into Eqn. (3.15),

$$\vartheta_{ik} = \frac{d}{\lambda_o} \cos \theta_i \sin \phi_k, \quad (3.63)$$

where  $\lambda_o$  is the nominal wavelength,  $\theta_i$  is the elevation angle to the patch, and  $\phi_k$  is the azimuth angle to the patch. The normalized Doppler frequency for the  $ik^{th}$  clutter patch is expressed as  $\bar{\omega}_{ik}$ .

Since the clutter is extended along the entire Earth's surface to the horizon, the unambiguous range of the radar becomes an important consideration in the development of the model. The radar's unambiguous range  $R_u$  is given by Skolnik [17, page 3] as

$$R_u = \frac{c}{2f_r}, \quad (3.64)$$

where  $c$  is the propagation velocity and  $f_r$  is the PRF. Assuming the radar's unambiguous range is less than the range to the radar horizon, the clutter return will consist of echoes from multiple

---

<sup>3</sup>The  $ik^{th}$  clutter patch is defined as the  $k^{th}$  patch on the  $i^{th}$  ring.

ranges. Therefore, each ambiguous range is given by

$$R_i = R_c + (i - 1)R_u \text{ for } i = 1, \dots, N_r. \quad (3.65)$$

The number of range ambiguities to the horizon is denoted  $N_r$ .

From these parameters, the component of the space-time snapshot due to clutter  $\chi_c$  is given by

$$\chi_c = \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \alpha_{ik} \mathbf{v}(\vartheta_{ik}, \bar{\omega}_{ik}) = \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \alpha_{ik} \mathbf{b}(\bar{\omega}_{ik}) \otimes \mathbf{a}(\vartheta_{ik}), \quad (3.66)$$

where  $\alpha_{ik}$  denotes the random amplitude of the  $ik^{th}$  clutter patch.

The next step is to determine the Clutter-to-Noise Ratio (CNR) for each patch. This step is similar to the derivation of the JNR. However, the Radar Cross Section (RCS) of each clutter patch must be determined because of the two way radar equation. The dependence of the RCS of the  $ik^{th}$  clutter patch is explicit to highlight the fact the aircraft is flying over a non-homogeneous surface and hence the RCS per unit area is dependent on the clutter patch under consideration. The RCS per unit area is given by Skolnik [17, page 471] as

$$\sigma_o(\theta_i, \phi_k) = \frac{\sigma_c(\theta_i, \phi_k)}{A_c}, \quad (3.67)$$

where  $\sigma_c$  is the RCS due to the area circumscribed by  $A_c$ . Using this relationship, the RCS of the  $ik^{th}$  clutter patch is given by

$$\sigma_{ik} = \sigma_o(\theta_i, \phi_k) \times A_{ik}, \quad (3.68)$$

where the  $A_{ik}$  is the area of the  $ik^{th}$  clutter patch (assumed constant for all  $i$  and  $k$ ). Using the patch area given by Skolnik [17, page 472], the RCS becomes

$$\begin{aligned}
 \sigma_{ik} &= \sigma_o(\theta_i, \phi_k) R_i \Delta\phi \frac{cT_p}{2} \sec \psi_i \\
 &= \sigma_o(\theta_i, \phi_k) R_i \Delta\phi \frac{c}{2B} \sec \psi_i \\
 &= \sigma_o(\theta_i, \phi_k) R_i \Delta\phi \Delta R \sec \psi_i.
 \end{aligned} \tag{3.69}$$

The data for land clutter is sometimes given by the parameter  $\gamma$ , related to the nominal RCS of the clutter patch [17, page 490] through the relationship

$$\sigma_o = \gamma \sin \psi_c, \tag{3.70}$$

where  $\psi_c$  is again the grazing angle. This model serves as the basis for the clutter simulations found in this chapter. The terrain dependent parameter is taken to be  $\gamma$ , however Skolnik states  $\gamma$  also depends on the grazing angle if the angle of incidence is near perpendicular or the grazing angle is extremely low in the case of a low flying aircraft [17, page 490].

Using the above definition for the RCS of each clutter patch, the CNR at the  $ik^{th}$  clutter patch is given by the two way radar range equation,

$$\xi_{ik} = \frac{P_t T_p G_t(\theta_i, \phi_k) g(\theta_i, \phi_k) \lambda_o^2 \sigma_{ik}}{(4\pi)^3 N_o L_s R_i^4}, \tag{3.71}$$

where the element pattern is represented by  $g(\theta_i, \phi_k)$  as before and the two-sided receiver noise power spectral density is represented by  $\frac{N_o}{2}$ , such that the noise power per element is  $\sigma^2 = \frac{N_o}{2} 2B = N_o/T_p$  (since the radar is assumed to have a separate receiver for each antenna element).

Due to the variability of the clutter, the returns from different patches are assumed uncorrelated as shown by

$$\mathcal{E} \{ \alpha_{ik} \alpha_{i'k'}^* \} = \sigma^2 \xi_{ik} \delta_{i-i'} \delta_{k-k'}. \quad (3.72)$$

Notice the element noise power multiplied by the CNR results in the power in the  $ik^{th}$  clutter patch as expected. Cross correlation terms are zero. Using this fact, the clutter space-time covariance matrix  $\mathbf{R}_c$  is derived as follows,

$$\begin{aligned} \mathbf{R}_c &= \mathcal{E} \{ \chi_c \chi_c^H \} \\ &= \mathcal{E} \left\{ \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \alpha_{ik} \mathbf{v}(\vartheta_{ik}, \bar{\omega}_{ik}) \sum_{i'=1}^{N_r} \sum_{k'=1}^{N_c} \alpha_{i'k'}^* \mathbf{v}(\vartheta_{i'k'}, \bar{\omega}_{i'k'})^H \right\} \\ &= \mathcal{E} \left\{ \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \sum_{i'=1}^{N_r} \sum_{k'=1}^{N_c} \alpha_{ik} \alpha_{i'k'}^* \mathbf{v}(\vartheta_{ik}, \bar{\omega}_{ik}) \mathbf{v}(\vartheta_{i'k'}, \bar{\omega}_{i'k'})^H \right\} \\ &= \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \sigma^2 \xi_{ik} \mathbf{v}(\vartheta_{ik}, \bar{\omega}_{ik}) \mathbf{v}(\vartheta_{i'k'}, \bar{\omega}_{i'k'})^H, \end{aligned} \quad (3.73)$$

where the final step uses the results from Eqn. (3.72). The clutter space-time covariance matrix can also be expressed using the Kronecker Product,

$$\mathbf{R}_c = \sigma^2 \sum_{i=1}^{N_r} \sum_{k=1}^{N_c} \xi_{ik} \mathbf{b}(\bar{\omega}_{ik}) \mathbf{b}(\bar{\omega}_{ik})^H \otimes \mathbf{a}(\vartheta_{ik}) \mathbf{a}(\vartheta_{ik})^H. \quad (3.74)$$

An even more compact notation is available through the definition of the following matrices, similar to that done for the jammer covariance matrix. Begin by defining the steering vector matrix with the assumption that no range ambiguities exist ( $N_r = 1$ ),

$$\mathbf{V}_c = [\mathbf{v}_1; \mathbf{v}_2; \cdots; \mathbf{v}_{N_c}]. \quad (3.75)$$



Then form the clutter power distribution into the diagonal matrix

$$\Xi_c = \sigma^2 \text{diag}([\xi_1, \xi_2, \dots, \xi_{N_c}]). \quad (3.76)$$

Use of these two definitions results in the following form for the clutter space-time covariance matrix,

$$\mathbf{R}_c = \mathbf{V}_c \Xi_c \mathbf{V}_c^H. \quad (3.77)$$

**3.3.3.1 Clutter Ridges.** The relationship between the clutter patch and the motion of the airborne platform gives rise to a dependence on Doppler frequency. This relationship produces what is characterized by Ward [21, page 24] as a clutter ridge. To see this, first revisit the spatial frequency of the  $ik^{th}$  clutter patch,

$$\vartheta_{ik} = \frac{d}{\lambda_o} \cos \theta_i \sin \phi_k. \quad (3.78)$$

The Doppler frequency of this patch is given through the relationship defined previously as

$$f_{ik} = \frac{2v_{ik}}{\lambda_o} = \frac{2\hat{\mathbf{k}}(\theta_i, \phi_k) \cdot \mathbf{v}_a}{\lambda_o}, \quad (3.79)$$

where  $v_{ik}$  is the velocity of the  $ik^{th}$  clutter patch relative to the airborne platform and  $\mathbf{v}_a$  is the aircraft velocity vector. If the aircraft is assumed traveling in the  $\hat{\mathbf{x}}$  direction, the Doppler frequency becomes

$$f_{ik} = \frac{2v_a}{\lambda_o} \cos \theta_i \sin \phi_k. \quad (3.80)$$

Rearranging the above equation highlights the expression for spatial frequency,

$$\begin{aligned} f_{ik} &= \frac{2dv_a}{d\lambda_o} \cos \theta_i \sin \phi_k \\ &= \frac{2v_a}{d} \vartheta_{ik}, \end{aligned} \quad (3.81)$$

where  $d$  is the interelement spacing of the antenna. Normalizing this expression to the PRF as before results in

$$\bar{\omega}_{ik} = \frac{f_{ik}}{f_r} = f_{ik} T_r = \frac{2v_a T_r}{d} \vartheta_{ik}, \quad (3.82)$$

where  $T_r$  is the PRI. This is the relationship between the normalized Doppler frequency and the velocity of the aircraft. Since the spatial frequency of the  $ik^{th}$  clutter patch is dependent only on  $\phi_k$  (the aircraft is assumed at a constant altitude), the normalized Doppler frequency is linearly dependent on the azimuth with a slope  $\beta$  given by

$$\beta = \frac{2v_a T_r}{d}. \quad (3.83)$$

Examining the units in this relationship shows it represents the number of half-interelement spacings  $d$  crossed during one PRI. If this slope is greater than one the clutter is Doppler ambiguous since it is spread over a frequency band wider than the PRF. If it is less than one it is unambiguous. To further understand this Doppler ambiguity relationship, examine the Fourier series of the receive clutter echo pulse.

The Fourier series representation of a rectangular pulse is a series of delta functions spaced a distance  $f_r$  apart (the PRF), the envelope defined by a sine over argument curve. The Doppler shift of the clutter patch return moves the entire spectrum either up or down in frequency, depending on whether the aircraft is approaching or receding from the patch. The receiver bandwidth  $B$  is

fixed about the carrier frequency and does not move in correlation with this Doppler shift. The bandwidth of the receiver is less than the PRF. Therefore, if the Doppler shift is enough to move the spectrum so that one of the outlying pulses is now in the receiver bandwidth, the radar cannot distinguish between it and the original PRF line. Hence, the Doppler frequency is ambiguous or aliased.

As  $\beta$  increases it becomes more difficult to suppress the clutter. The aliasing in the Doppler frequency space explains this relationship. Increased aliasing means more clutter returns contain the same Doppler frequency as the target, further obscuring it. Figure 3.18 shows a plot of an adapted antenna pattern where clutter ridge null is easily visible for  $\beta = 1$ . From the definition of  $\beta$ , it is obvious the velocity of the aircraft plays a role in clutter suppression. Figure 3.7 shows several plots illustrating the aliasing of the normalized Doppler frequency for various values of  $\beta$ .

In the  $\beta = 1$  plot only one Doppler frequency exists for every possible spatial frequency. Obviously, for values of  $\beta$  less than one this relationship still exists. This is termed the unambiguous Doppler condition. As  $\beta$  is increased more and more spatial frequencies can be assigned to any one Doppler frequency. Hence, the ambiguous Doppler condition. Figure 3.7c shows a case where as many as four spatial frequencies can be associated with any one Doppler frequency.

**3.3.3.2 Rank of the Clutter Space-Time Covariance Matrix.** Because the clutter ridge is linear, it suggests the rank of the clutter matrix is low. This is supported by Brennan's rule. This rule states

$$\text{rank}(\mathbf{R}_c) \approx \lfloor N + (M - 1)\beta \rfloor, \quad (3.84)$$

where the  $\lfloor \cdot \rfloor$  symbols indicate rounding to the nearest integer. Brennan's rule states the rank of the clutter covariance matrix is a function of the number of antenna elements, the CPI length, and

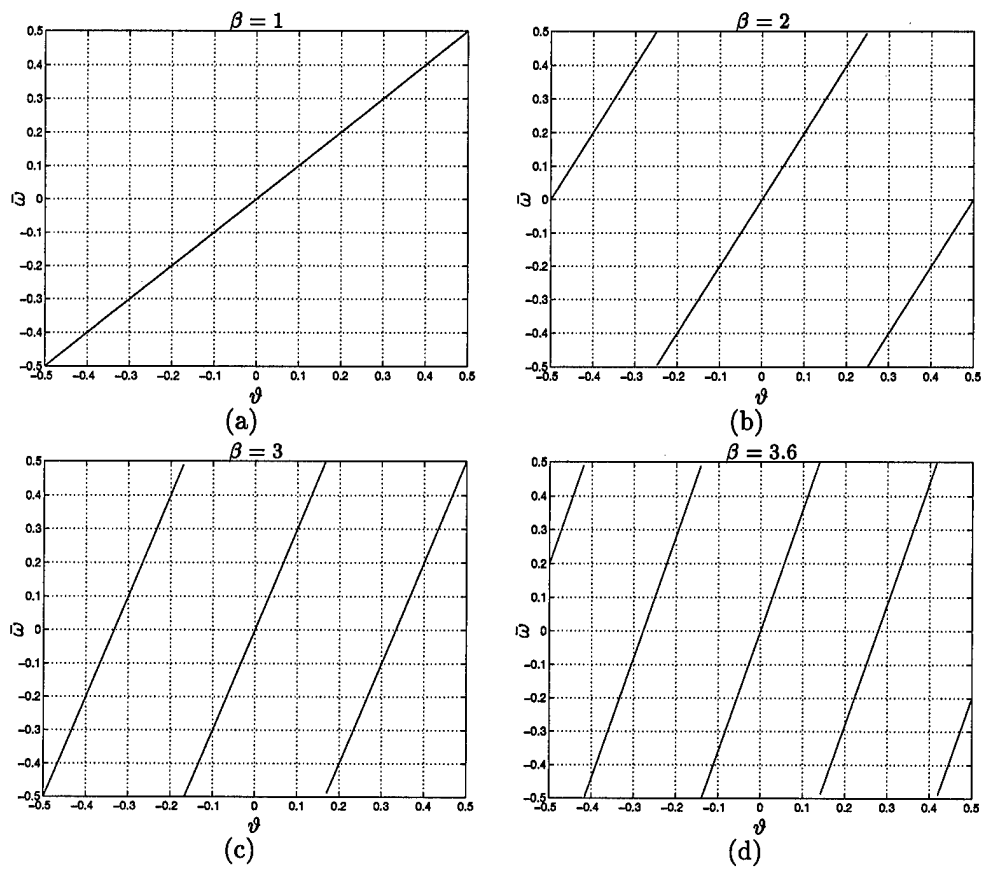


Figure 3.7 Clutter ridge plots illustrating the normalized Doppler aliasing for various values of  $\beta$ .

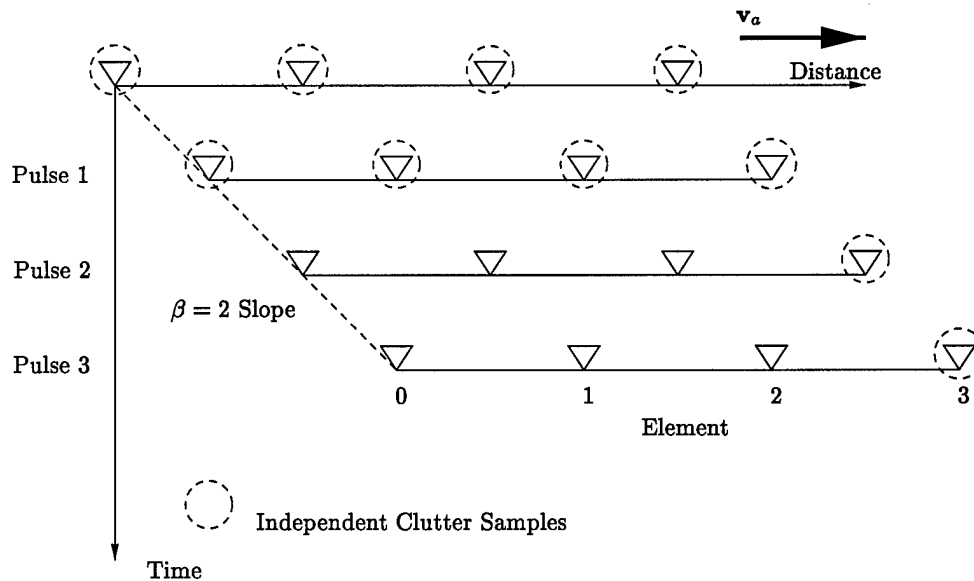


Figure 3.8 Correlated clutter sampling ( $\beta$  equal to an integer) gives low rank.

the slope of the clutter ridge. For integer values of  $\beta$  this is an accurate estimate of the rank as Fig. 3.11 shows for the upcoming example.

Brennan's rule makes a lot sense. It is primarily a function of  $\beta$ , the number of half-interelement spacings transcended during one PRI. If  $\beta$  is an integer the antenna elements basically take the same clutter sample as the aircraft travels. Since these samples correspond to the same spatial position, the clutter space-time covariance matrix rank is low due to the strong correlation between samples (they are not linearly independent). Figure 3.8 illustrates this concept.

**3.3.3.3 CNR Plot and Clutter Covariance Matrix Rank Examples.** Given the element voltage pattern described below and the scenario listed in Table 3.1, let us calculate the resulting Clutter-to-Noise Ratio (CNR) pattern for the  $ik^{th}$  clutter patch and also the rank of the clutter space-time covariance matrix for various values of  $\beta$ . The antenna array is rectangular with elements in the  $x$  and  $z$  directions. The beamforming operation uses only a row of elements in the

Table 3.1 Scenario parameters for the clutter example.

Variable	Value
$M$	18
$N$	18
$f_o$	450 MHz
$f_r$	300 Hz
$T_p$	200 $\mu s$
$P_t$	200 kW
$B$	4 MHz
$F_n$ (Noise Figure)	3 dB
$N_c$	361
$h_a$	9 km
$R_c$	130 km
$\gamma$	-3 dB
Transmit Gain	22 dB
Column Receive Gain	10 dB
Number of Elements $x$	18
Number of Elements $z$	4
Element Pattern	Cosine
Element Gain	4 dB
Element Backlobe Level	-30 dB
Transmit Taper	Uniform (None)

$x$  direction, 18 elements, each with the voltage pattern

$$f(\theta_i, \phi_k) = \begin{cases} \cos \phi_k & -90^\circ \leq \phi_k < 90^\circ \text{ frontlobe} \\ b_e \cos \phi_k & 90^\circ \leq \phi_k < 270^\circ \text{ backlobe.} \end{cases}$$

The value of  $b_e$  required to give a backlobe of -30 dB is 0.032, the inverse of the dB operation.

Figure 3.9 shows the antenna's pattern using this value.

The transmit gain in Eqn. (3.71) is a function of the azimuth and elevation angles as given by the array pattern. The array pattern is calculated using the same techniques as in Chapter II and is given by

$$W(\mathbf{k} - \mathbf{k}_o) = \frac{1}{N} \sum_{i=0}^{N-1} w_i e^{-j(\mathbf{k} - \mathbf{k}_o) \cdot \mathbf{x}_i},$$

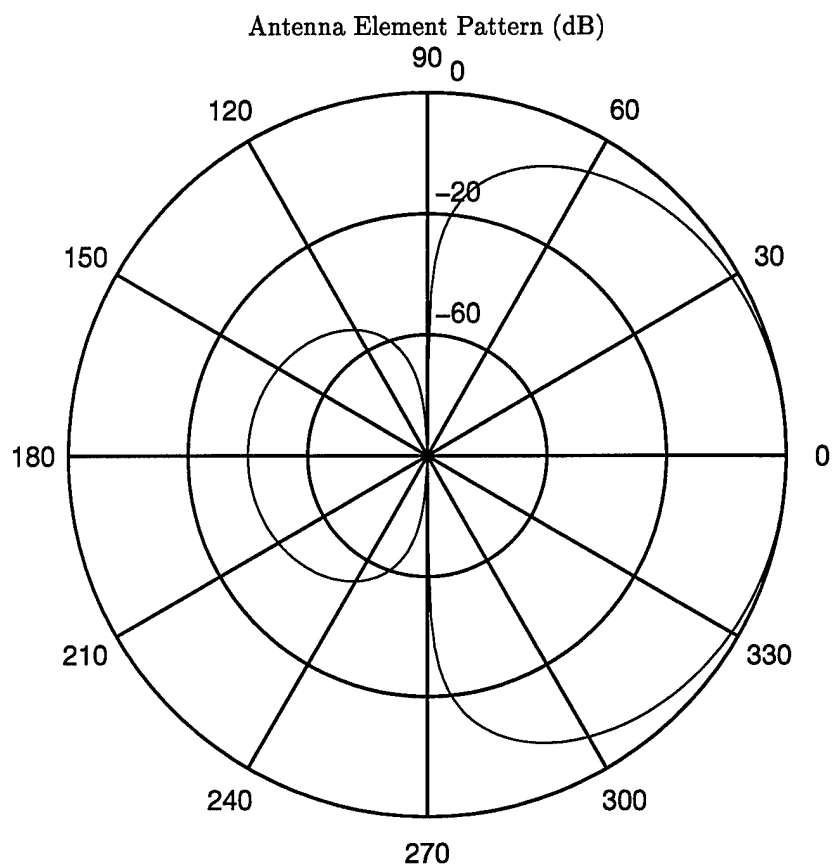


Figure 3.9 Element voltage pattern for the clutter example.

where  $\mathbf{k}$  is the spatial frequency vector and  $w_i$  is the individual antenna element weight. No weights have been applied to the antenna element signals, hence  $w$  is equal to one for all elements. Also, the antenna beam is not steered in this example so  $\mathbf{k}_o$  is equal to the zero vector, resulting in a beam along the boresight of the array. Using Eqn. (2.3), the spatial frequency is given by

$$\begin{aligned}\mathbf{k} &= \frac{2\pi}{\lambda} \hat{\mathbf{k}}(\theta_i, \phi_k) \\ &= \frac{2\pi}{\lambda_o} \{\cos \theta_i \sin \phi_k \hat{x} + \cos \theta_i \cos \phi_k \hat{y} + \sin \theta_i \hat{z}\}.\end{aligned}$$

Notice, given the antenna elements oriented in the  $\hat{x}$  direction, the formula coincides with the expression for spatial frequency found in Eqn. (3.15). Since the beamforming operation is only across the elements in the  $\hat{x}$  direction, the outer product in the exponential simplifies the equation to

$$W(\theta_i, \phi_k) = \frac{1}{N} \sum_{n=0}^{N-1} e^{-j \frac{2\pi n d}{\lambda_o} \cos \theta_i \sin \phi_k}.$$

The summation's argument can be written to the power of it's index and simplified,

$$\begin{aligned}W(\theta_i, \phi_k) &= \frac{1}{N} \sum_{n=0}^{N-1} \left( e^{-j \frac{2\pi d}{\lambda_o} \cos \theta_i \sin \phi_k} \right)^n \\ &= \frac{1}{N} e^{-j \frac{\pi d(N-1)}{\lambda_o} \cos \theta_i \sin \phi_k} \frac{\sin \left( \frac{\pi d N}{\lambda_o} \cos \theta_i \sin \phi_k \right)}{\sin \left( \frac{\pi d}{\lambda_o} \cos \theta_i \sin \phi_k \right)}.\end{aligned}$$

The transmit gain pattern for the array is now given by (the antenna pattern as derived in Chapter II is voltage)

$$G_t(\theta_i, \phi_k) = g(\theta_i, \phi_k) W^2(\theta_i, \phi_k),$$



where the element power pattern  $g(\theta_i, \phi_k)$  is simply the voltage pattern squared,  $|f(\theta_i, \phi_k)|^2$ .

The next step is to determine the elevation angle for the given scenario. Using Eqn. (3.61) and the values given in Table 3.1, the elevation angle is calculated as -0.0769 radians. The cosine of this angle is approximately one and is ignored from this point forward.

The next major calculation is the determination of the RCS from the  $ik^{th}$  clutter patch. To begin, determine the grazing angle for the aircraft's given altitude and range of interest. Substituting the appropriate values into Eqn. (3.62) shows the grazing angle to be 0.0617 radians. From here, the RCS per unit area is easily calculated given the terrain parameter  $\gamma$ ,

$$\sigma_o = 10^{-0.3} \sin(0.0617) = 0.0309.$$

The granularity of the patches around the clutter ring of interest is given by

$$\Delta\phi = \frac{2\pi}{N_c} = \frac{2\pi}{361} = 0.0174 \text{ radians}$$

and the radar's range resolution is

$$\Delta R = \frac{c}{2B} = \frac{3 \times 10^8}{2(4 \times 10^6)} = 37.5 \text{ m.}$$

This example is performed at the first ambiguous range, therefore  $\psi_i = \psi_c$ . Also, our interest is focused on only one clutter ring. Using these assumptions and the constant  $\gamma$  clutter model, the RCS of the  $ik^{th}$  clutter patch is constant around the clutter ring and equal to

$$\begin{aligned} \sigma_{ik} &= \sigma_o R_i \Delta\phi \Delta R \sec \psi_i \\ &= 0.0309 \times (130 \times 10^3) \times 0.0174 \times 37.5 \times \sec(0.0617) \\ &= 2626.09 \text{ m}^2. \end{aligned}$$

The last unknown parameter for the CNR pattern (Eqn. (3.71)) is the receiver noise power spectral density,  $N_o$ . The system noise figure  $F_n$  is given in the table. Therefore, the noise power spectral density in the receiver is [17, page 19]

$$F_n = \frac{N_o}{kT_o G_r} \text{ or } N_o = F_n kT_o G_r,$$

where the gain of the receiver  $G_r$  is assumed equal to one, the system noise figure  $F_n$  is given as part of the problem,  $k$  is Boltzman's constant, and  $T_o$  is the standard temperature (290°K). Solving for  $N_o$  gives

$$N_o = 10^{0.3} \times (13.80658 \times 10^{-24}) \times 290 = 319.554 \times 10^{-15}.$$

This is the noise due to thermal activity in the receiver.

Substituting these values and an interelement spacing  $d$  equal to half the wavelength into Eqn. (3.71) results in the plot shown in Fig. 3.10. This is the CNR around the entire clutter ring. Since the antenna beam is not steered, the main beam is located at zero degrees azimuth. Notice the sharp roll-off at  $\pm 90^\circ$  azimuth due to the antenna element patterns. These locations represent the ends of the antenna array.

The next part of the example examines the performance of Brennan's rule in determining the rank of the clutter space-time covariance matrix  $\mathbf{R}_c$ . Since we want to plot the rank of the clutter matrix for various values of  $\beta$ , first determine the Doppler frequency of the  $ik^{th}$  clutter patch in terms of this using Eqn. (3.83),

$$\begin{aligned} f_{ik} &= \frac{2v_a}{\lambda_o} \cos \theta_i \sin \phi_k \\ &= \frac{2\beta d}{2T_r \lambda_o} \cos \theta_i \sin \phi_k \\ &= \frac{\beta d}{T_r \lambda_o} \cos \theta_i \sin \phi_k. \end{aligned}$$

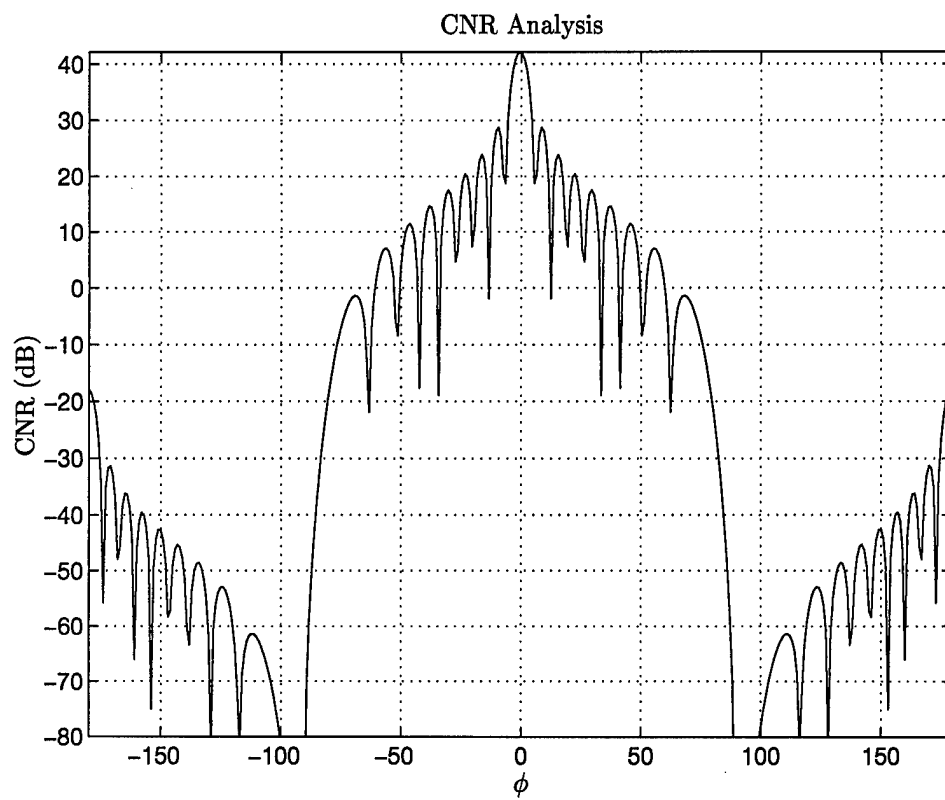


Figure 3.10 CNR around the clutter ring (main beam pointed at 0° azimuth).

The temporal steering vector requires the normalized Doppler of the  $ik^{th}$  clutter patch (see Eqn. (3.27)) given by

$$\bar{\omega}_{ik} = \frac{f_{ik}}{f_r} = \frac{\beta d}{\lambda_o} \cos \theta_i \sin \phi_k.$$

From this point on, the formulation of the clutter matrix is the implementation of Eqn. (3.74) in MATLAB<sup>®</sup>. Because of the peculiarities of MATLAB<sup>®</sup> the implementation is not discussed here. The element noise power  $\sigma^2$  is assumed equal to one for this simulation.

Figure 3.11 shows the resulting ranks for clutter matrices corresponding to  $\beta$  values of 0.6, 1, 2, 2.83, and 3. Because the number of nonzero eigenvalues is synonymous with the rank of a matrix, this metric is used. As mentioned previously, Brennan's rule predicts the rank of the matrix very well when  $\beta$  is an integer. This means the number of half-interelement spaces transcended during a PRI is an integer value. Therefore the antenna elements line up and the subsequent elements basically retrieve the same clutter sample as the first. Hence, the clutter matrix contains correlated clutter samples. This condition is shown in Fig. 3.8. However, Brennan's rule does not hold when given uncorrelated clutter samples (non-integer values of  $\beta$ ) as shown by the tapering eigenvalues for the  $\beta = 0.6$  and  $\beta = 2.83$  curves.

### 3.4 Direct Form Fully Adaptive STAP

The purpose of Space-Time Adaptive processing is to direct an antenna null in the direction of a jammer or a significant clutter source. This minimizes the effect of the interference and can lead to the detection of a target in environments standard radar processing cannot handle. This section discusses computation of the antenna pattern/filter weights to steer the antenna nulls using a fully adaptive algorithm.

The two basic approaches to STAP are Fully Adaptive STAP and Partially Adaptive, or Reduced-Dimension STAP. There are inherent advantages and disadvantages to both. Obviously,

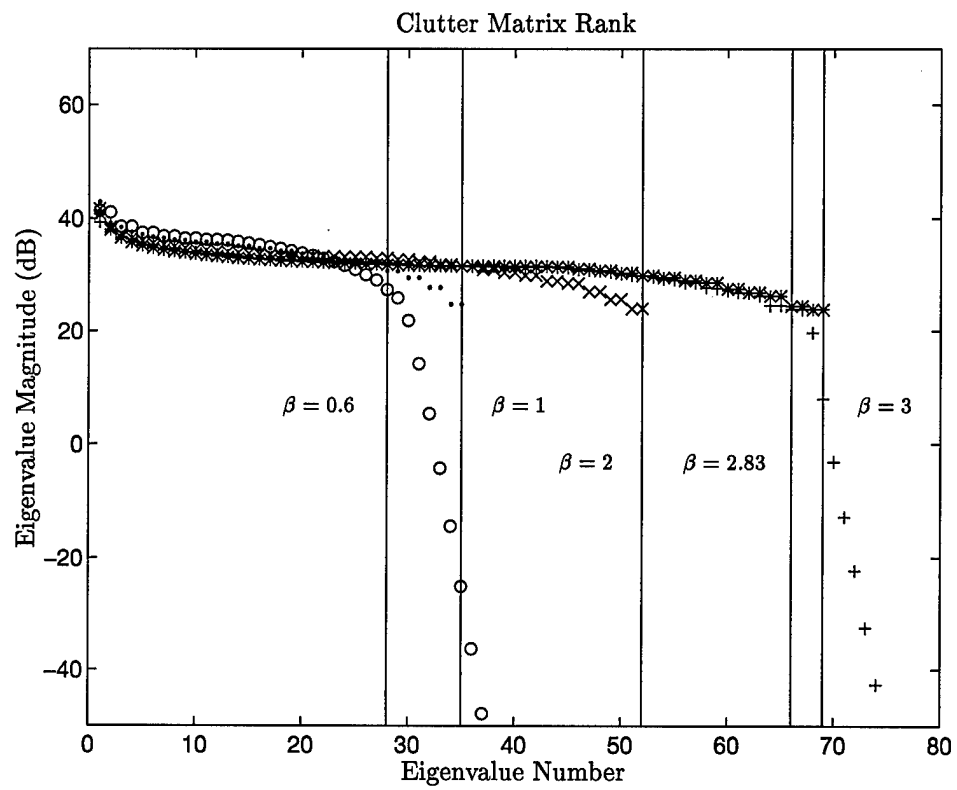


Figure 3.11 Illustration of Brennan's rule for clutter matrix rank using various values of  $\beta$ .

the best performance to be obtained is with a fully adaptive algorithm. However, the computational burden becomes too much to perform within the timeline of the typical radar.

Fully adaptive STAP results in a weight vector for every element of the space-time snapshot (see Eqn. (3.31)), i.e. every element and pulse in the CPI is weighted by the digital processor. This entire process is described by

$$z = \mathbf{w}^H \boldsymbol{\chi}. \quad (3.85)$$

The vector  $\mathbf{w}$  is the *weight vector* defined by [21]

$$\mathbf{w} = \mathbf{R}^{-1} \mathbf{v}_t, \quad (3.86)$$

where  $\mathbf{R}$  is defined as the sum of all the undesired component space-time covariance matrices,  $\mathbf{v}_t$  is the target steering vector, and  $z$  is the output of the algorithm. The undesired space-time covariance matrix is given by

$$\mathbf{R} = \mathbf{R}_n + \mathbf{R}_j + \mathbf{R}_c, \quad (3.87)$$

where  $\mathbf{R}_n$  represents thermal noise,  $\mathbf{R}_j$  represents jammers, and  $\mathbf{R}_c$  represents clutter.

The expression for the algorithm (Eqn. (3.85)) maximizes Signal-to-Interference-plus-Noise-Ratio (SINR), maximizes probability of detection for a given false alarm probability, and with the proper choice of scale factor minimizes the output noise power subject to a unity gain constraint in the target direction [21, page 57]. One problem with this approach is high sidelobes in both angle and Doppler. For this reason Ward also introduces the tapered weight vector,

$$\mathbf{g}_t = \mathbf{t} \odot \mathbf{v}_t, \quad (3.88)$$

where  $\mathbf{t}$  is a two dimensional window/taper and  $\odot$  is the Hadamard product (equivalent to an element by element multiply). This two dimensional window can be generated from two one dimensional windows using the Kronecker product as shown by

$$\mathbf{t} = \mathbf{t}_b \otimes \mathbf{t}_a, \quad (3.89)$$

where  $\mathbf{t}_b$  and  $\mathbf{t}_a$  are the one dimensional window functions. An expression equivalent to Eqn. (3.88) is

$$\mathbf{g}_t = (\mathbf{t}_b \odot \mathbf{b}_t) \otimes (\mathbf{t}_a \odot \mathbf{a}_t). \quad (3.90)$$

This tapered space-time steering vector can be substituted for  $\mathbf{v}_t$  in Eqn. (3.86). The window in this new weight vector produces low sidelobes at the sacrifice of mainbeam width and gain. Correspondingly, it no longer produces the optimum SINR. Harris [11] offers an excellent discussion on windowing with a direct comparison of the different types of windows. The SINR is defined as [21]

$$\text{SINR} = \frac{\sigma^2 \xi_t |\mathbf{w}^H \mathbf{v}_t|^2}{\mathbf{w}^H \mathbf{R} \mathbf{w}}, \quad (3.91)$$

where the numerator represents the signal power and the denominator represents the interference plus noise power. To get the optimum SINR, substitute the optimum weight vector  $\mathbf{w}$  from Eqn. (3.86) into the above expression and simplify,

$$\begin{aligned} \text{SINR}_{\text{opt}} &= \frac{\sigma^2 \xi_t |\mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t|^2}{\mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{R} \mathbf{R}^{-1} \mathbf{v}_t} \\ &= \frac{\sigma^2 \xi_t \mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t \mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t}{\mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t} \\ &= \sigma^2 \xi_t \mathbf{v}_t^H \mathbf{R}^{-1} \mathbf{v}_t, \end{aligned} \quad (3.92)$$

where  $\sigma^2$  is the noise power per element and  $\xi_t$  is the target SNR per element. Since Eqn. (3.92) is for one particular target azimuth and Doppler, it can be expanded across normalized Doppler by highlighting the temporal steering vector dependence on  $\bar{\omega}$ ,

$$\text{SINR}_{\text{opt}}(\bar{\omega}) = \sigma^2 \xi_t (\mathbf{b}_t(\bar{\omega}) \otimes \mathbf{a}_t)^H \mathbf{R}^{-1} (\mathbf{b}_t(\bar{\omega}) \otimes \mathbf{a}_t). \quad (3.93)$$

The equation could also be written to highlight the dependence on spatial frequency because of the spatial steering vector but then the transmit and receive antenna patterns have to be taken into account when plotting this expression.

Similarly, the SINR for the tapered pattern can be obtained by substituting Eqn. (3.88) into Eqn. (3.91) and simplifying,

$$\text{SINR}_{\text{tap}} = \frac{\sigma^2 \xi_t |\mathbf{g}_t^H \mathbf{R}^{-1} \mathbf{v}_t|^2}{\mathbf{g}_t^H \mathbf{R}^{-1} \mathbf{g}_t}. \quad (3.94)$$

Highlighting the dependence on normalized Doppler  $\bar{\omega}$  produces

$$\text{SINR}_{\text{tap}}(\bar{\omega}) = \frac{\sigma^2 \xi_t |[(\mathbf{t}_b \odot \mathbf{b}_t(\bar{\omega})) \otimes (\mathbf{t}_a \odot \mathbf{a}_t)]^H \mathbf{R}^{-1} (\mathbf{b}_t(\bar{\omega}) \otimes \mathbf{a}_t)|^2}{[(\mathbf{t}_b \odot \mathbf{b}_t(\bar{\omega})) \otimes (\mathbf{t}_a \odot \mathbf{a}_t)]^H \mathbf{R}^{-1} [(\mathbf{t}_b \odot \mathbf{b}_t(\bar{\omega})) \otimes (\mathbf{t}_a \odot \mathbf{a}_t)]}. \quad (3.95)$$

Figure 3.12 shows the effect of three different types of windows on the SINR for a fully adaptive processor. The plots shown are for an 18 azimuth element array with 18 pulses in the CPI. The noise power per element  $\sigma^2$  and target SNR per element per pulse  $\xi_t$  are both set to one for the plots. The interference matrix  $\mathbf{R}$  is the same as that generated for the previous clutter example with the addition of two jammers and noise. The windows/tapers were applied in both the spatial and temporal domains. Referencing Fig. 3.12 shows the poorest performance of the three tapers shown is with the Blackman Harris window. This poor performance is expected because of the extremely low -90 dB sidelobes produced by this window in both the spatial and temporal



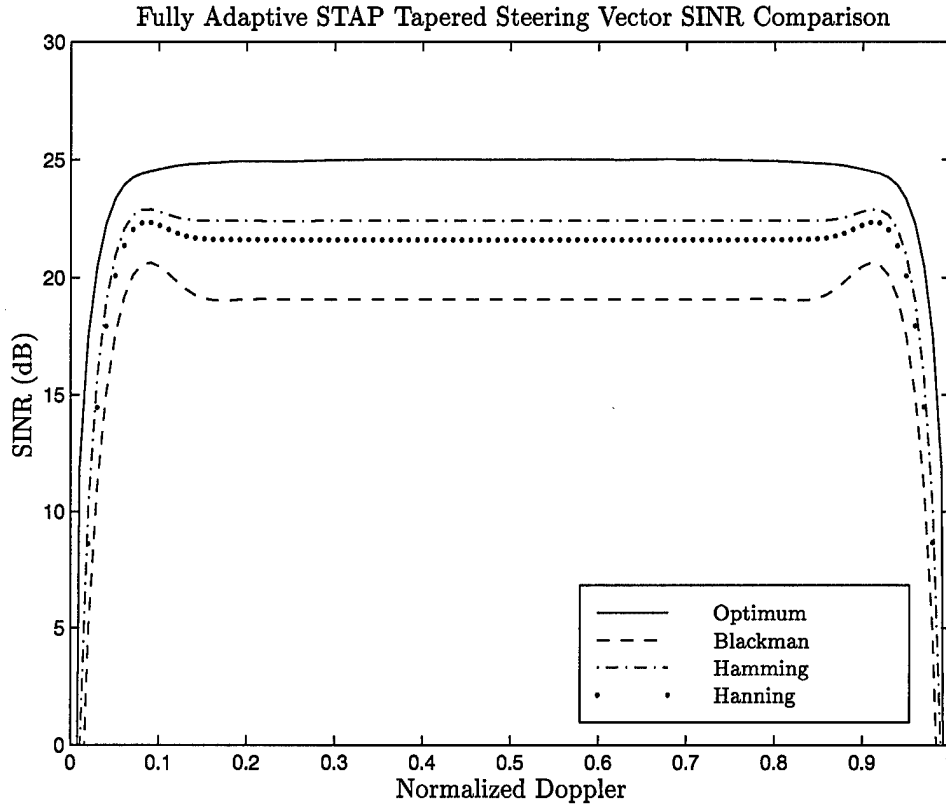


Figure 3.12 An SINR comparison of fully adaptive STAP for the Blackman Harris, Hanning, and Hamming [11] tapered steering vectors.

domains. The loss in main beam gain outweighs the gain found by suppressing the interference with the extremely low sidelobes.

The SINR represents a good tool for the comparison of various STAP algorithms. Another tool is the SINR loss. This metric compares the performance of the algorithm to the optimal case, a noise only environment. In the noise only environment, the interference covariance matrix becomes

$$\mathbf{R} = \mathbf{R}_n = \sigma^2 \mathbf{I}_{MN}. \quad (3.96)$$

Substituting this noise only covariance matrix into Eqn. (3.92) results in the optimum Signal-to-Noise Ratio (SNR),

$$\text{SNR}_{\text{opt}} = \sigma^2 \xi_t \mathbf{v}_t^H \frac{1}{\sigma^2} \mathbf{I}_{MN} \mathbf{v}_t = \xi_t \mathbf{v}_t^H \mathbf{v}_t. \quad (3.97)$$

Since the space-time steering vector is the complex exponential form of Eqns. (3.27) and (3.26), it's inner product is simply the number of elements in the vector, or  $MN$ . Therefore, the optimum SNR becomes

$$\text{SNR}_{\text{opt}} = MN \xi_t, \quad (3.98)$$

where  $\xi_t$  is the target SNR per element per pulse. Using this optimum SNR, we can define the SINR loss of the space-time processing algorithm as

$$L_{\text{SINR}} = \frac{\text{SINR}(\bar{\omega})}{\text{SNR}_{\text{opt}}}. \quad (3.99)$$

A plot of this metric corresponding to the windows used in Fig. 3.12 is shown in Fig. 3.13.

A block diagram of the basic STAP process is shown in Fig. 3.14. The process involves three major steps:

- Train the processor, i.e. generate a second set of data for use in the computation of the filter weights. The entire CPI datacube is used for the training algorithm in fully adaptive STAP.
- Generate the weights using the new set of data obtained from step one.
- Apply the weights to the space-time snapshot at the range gate of interest.

Once the weights have been applied, the result is sent to the radar threshold detection circuit to make a target decision. A more detailed diagram of the fully adaptive STAP process is shown in Fig. 3.15. The enormous amount of data calculation is more readily apparent from this diagram.

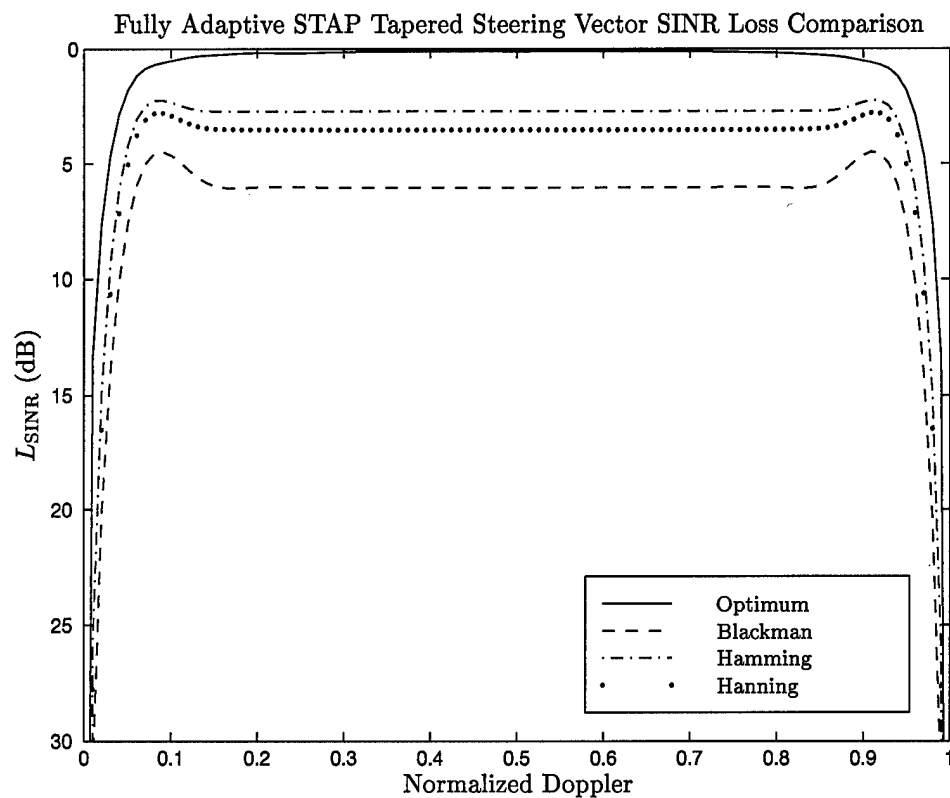


Figure 3.13 An SINR loss comparison of fully adaptive STAP for the Blackman Harris, Hanning, and Hamming [11] tapered steering vectors.

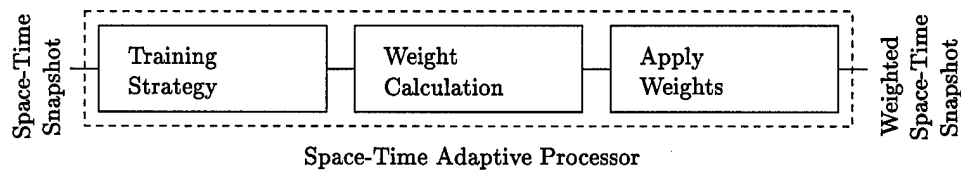


Figure 3.14 Top level block diagram for a space-time adaptive processor.

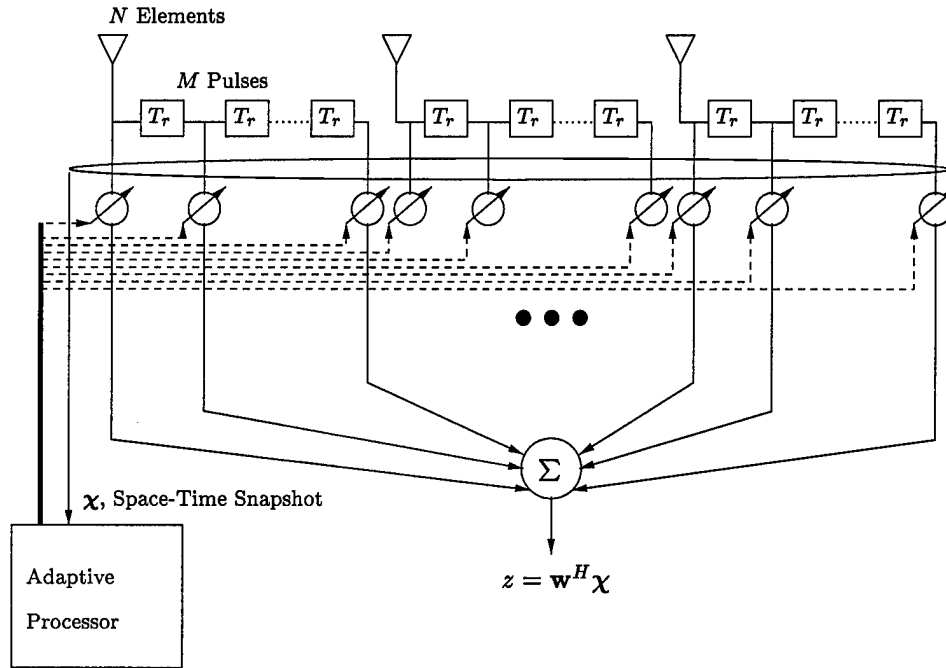


Figure 3.15 Fully adaptive space-time processing model.

Another metric of a STAP algorithm's performance is the antenna pattern it produces. The adapted pattern  $P_{\mathbf{w}}(\vartheta, \bar{\omega})$  is given by [21]

$$P_{\mathbf{w}}(\vartheta, \bar{\omega}) = |\mathbf{w}^H \mathbf{v}(\vartheta, \bar{\omega})|^2, \quad (3.100)$$

where  $\mathbf{w}$  is the weight vector and  $\mathbf{v}$  is the steering vector (not necessarily steered in the direction of the target).

An example of this adapted pattern using the parameters from the clutter example is shown in Figs. 3.16, 3.17, and 3.18. These patterns also show the introduction of two jammers, both at the same elevation as the airborne platform. The azimuth locations of the jammers are  $-40^\circ$  and  $25^\circ$  at a range of 370 km. The effective radiated power density for each jammer is 1 kW/MHz. The target is located at  $0^\circ$  azimuth with a Doppler frequency of 100 Hz. The value of  $\beta$  for the plots is 1, reflected by the clutter ridge null shown in the fully adapted pattern found in Fig. 3.18.

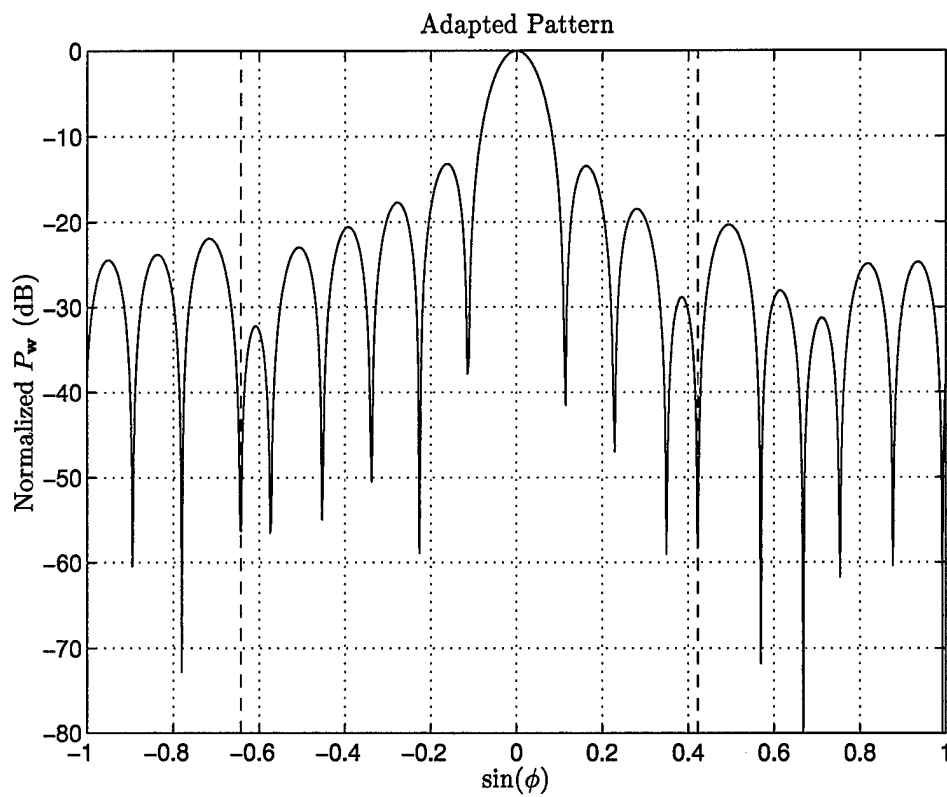


Figure 3.16 Fully adapted antenna pattern at 100Hz Doppler frequency. Vertical lines indicate jammer locations.

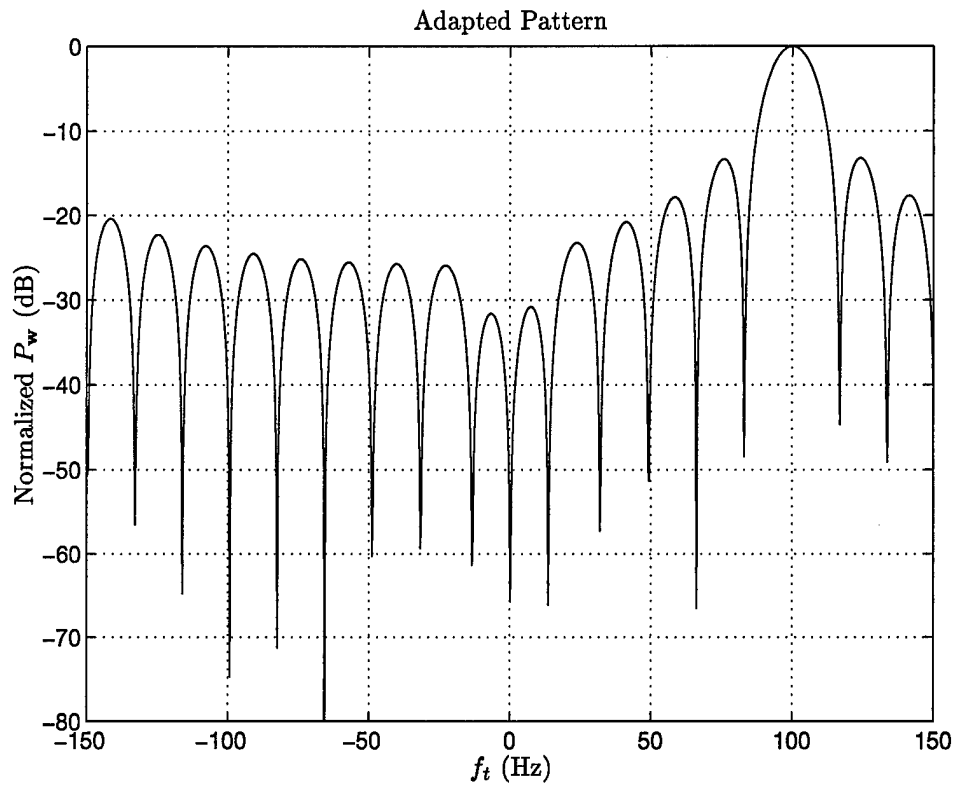


Figure 3.17 Fully adapted antenna pattern at  $0^\circ$  azimuth.

Figure 3.16 shows a cut along the target Doppler from Fig. 3.18. Since the target is located at  $0^\circ$  azimuth, the mainbeam of the pattern is located at  $0^\circ$  azimuth. Notice the nulls placed along the jammer locations ( $\sin(-40^\circ) = -0.643$  and  $\sin(25^\circ) = 0.423$ ). The nulls actually extend much lower. The plot doesn't show this because of the granularity of the azimuth angles used to generate it. Calculations at these exact azimuth values show the nulls possess greater than 90 dB attenuation.

A cut along the target's azimuth is shown in Fig. 3.17. The mainbeam is placed at the target's Doppler at 100 Hz with a strong null at zero Doppler. Zero Doppler corresponds to mainbeam clutter, a significant source of target acquisition difficulties. Again, the granularity of the Doppler frequency values used to generate the plot does not show the true depth of the null. A calculation at exactly zero Doppler shows this null is greater than 90 dB down as in the case of the jammers.

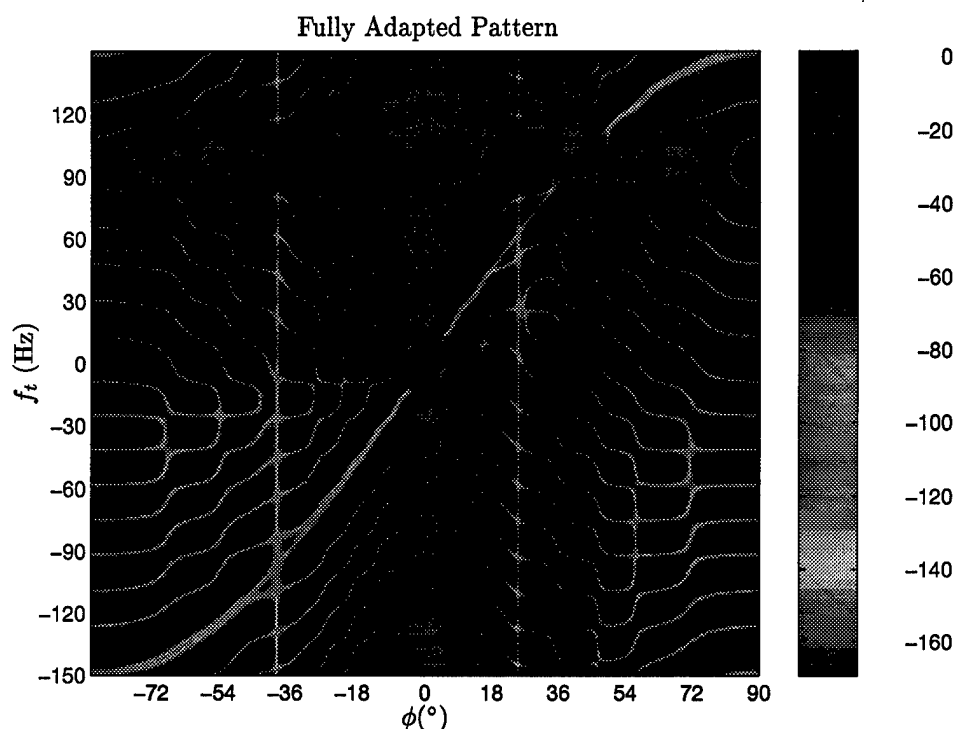


Figure 3.18 Three dimensional image of the fully adapted pattern  $P_w$  with no tapering for a single target and two jammers.

The final plot (Fig. 3.18) shows the entire pattern as viewed from the surface. The color represents the amplitude of the lobes. This figure shows the clutter ridge, corresponding to  $\beta = 1$ . Notice the ridge corresponds to the one predicted by Fig. 3.7c. It is not linear in this plot because it is plotted against  $\phi$  instead of  $\sin \phi$ .

Fully adaptive STAP is too computationally intensive to implement real time. The number of computations is on the order of  $(MN)^3$  [21]. Real time calculation is beyond current technology, although the computer industry is improving processor speed in such large steps that the ability to implement this algorithm could be reached in the near future (possibly within the next few years). Fully adaptive STAP is used primarily as the measure to compare reduced dimension algorithms. Reduced dimension algorithms can be implemented within the constraints of current computational technology.

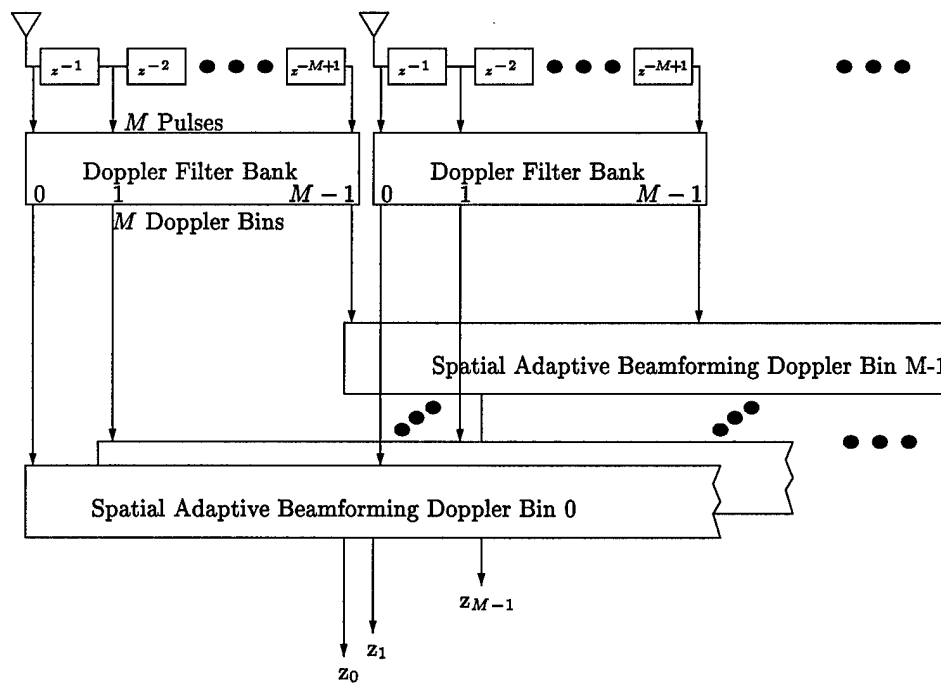


Figure 3.19 The Factored Time Space (FTS) reduced dimension algorithm block diagram. This diagram applies to one range gate and is repeated for every range gate of interest.

### 3.5 Partially Adaptive Factored Time-Space STAP

As just discussed, computational load drives the need for a reduced dimension algorithm. One of the simplest methods is the Factored Time-Space (FTS) algorithm.

This algorithm reduces the computational load by sacrificing the Doppler Degrees Of Freedom (DOF). The incoming signal from each antenna element is first Doppler filtered using the pulse returns from the range gate/cell of interest. A noise covariance matrix is then estimated from the Doppler filtered data and used to adaptively filter in the spatial domain, resulting in a scalar output for each Doppler bin. A block diagram of this process is shown in Fig. 3.19. The operations shown in this block diagram are repeated for each range cell of interest.

Although Fig. 3.19 shows a Doppler filter bank of  $M$  bins, equal to the number of pulses in the CPI, the number of bins does not in general have to be equal to the number of pulses. The Doppler filter is implemented with the Discrete Fourier Transform (DFT). The use of the DFT



allows us to zero pad the incoming data stream (pulse returns from a particular antenna element) to obtain a greater number of Doppler bins.

Using the DFT matrix allows us to mathematically express the FTS algorithm. Since this algorithm results in a single scalar for each Doppler bin, let's consider the mathematical architecture behind just a single Doppler bin,  $m$ . Begin by first obtaining the Doppler filter bank output for this bin. First define the Doppler filter bank using the  $M \times M$  DFT matrix  $\mathbf{W}$ . Since we have sacrificed all the DOF in the Doppler domain, we need a Doppler filter bank with very low sidelobes to suppress clutter. Therefore, let us express this low sidelobe Doppler filter bank as

$$\mathbf{F} = \text{diag}(\mathbf{t}_b) \mathbf{W}^H, \quad (3.101)$$

where  $\mathbf{t}_b$  is a window function (see Harris [11] for a complete discussion of windowing) and  $\text{diag}$  is the MATLAB<sup>®</sup> function for creating a diagonal matrix from the vector argument. The diagonal elements of the matrix are the elements of  $\mathbf{t}_b$ . Breaking this windowed DFT matrix down into its columns,

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_0 & \mathbf{f}_1 & \dots & \mathbf{f}_{M-1} \end{bmatrix}, \quad (3.102)$$

allows us to write the Doppler filter bank output for a particular bin. This output for Doppler bin  $m$  becomes

$$\tilde{\chi}_m = (\mathbf{f}_m \otimes \mathbf{I}_N)^H \chi, \quad (3.103)$$

where  $\chi$  is the space-time snapshot from Eqn. (3.29). The spatially adaptive filter weights for this Doppler bin become

$$\tilde{\mathbf{w}}_m = \tilde{\mathbf{R}}_m^{-1} \mathbf{a}_t, \quad (3.104)$$

where  $\mathbf{a}_t$  is the spatial steering vector as defined in Eqn. (3.26) (or a tapered spatial steering vector  $\tilde{\mathbf{g}}_t$  could be substituted) and the covariance matrix is given by

$$\tilde{\mathbf{R}}_m = \mathcal{E} \{ \tilde{\chi}_m \tilde{\chi}_m^H \} = (\mathbf{f}_m \otimes \mathbf{I}_N)^H \mathbf{R} (\mathbf{f}_m \otimes \mathbf{I}_N). \quad (3.105)$$

Using these weights, the output of the FTS algorithm for the  $m^{th}$  Doppler bin  $z_m$  is

$$z_m = \tilde{\mathbf{w}}_m^H \tilde{\chi}_m. \quad (3.106)$$

If we substitute Eqn. (3.103) into Eqn. (3.106), an expression results using the full dimensioned incoming data vector or space-time snapshot  $\chi$ ,

$$z_m = \tilde{\mathbf{w}}_m^H (\mathbf{f}_m \otimes \mathbf{I}_N)^H \chi. \quad (3.107)$$

A full dimension weight vector associated with the above equation can be defined as

$$\mathbf{w}_m^H = \tilde{\mathbf{w}}_m^H (\mathbf{f}_m \otimes \mathbf{I}_N)^H \quad (3.108)$$

or

$$\mathbf{w}_m = (\mathbf{f}_m \otimes \mathbf{I}_N) \tilde{\mathbf{w}}_m = \mathbf{f}_m \otimes \tilde{\mathbf{w}}_m. \quad (3.109)$$

This gives the alternate form of the adaptive processor's output,

$$z_m = \mathbf{w}_m^H \chi. \quad (3.110)$$

We can use the SINR as a comparison of the FTS algorithm to fully adaptive STAP. Substituting Eqn. (3.109) into Eqn. (3.91) results in an SINR for Doppler bin  $m$ ,

$$\text{SINR}_m = \frac{\sigma^2 \xi_t \left| (\mathbf{f}_m \otimes \tilde{\mathbf{w}}_m)^H (\mathbf{b}_t(\bar{\omega}) \otimes \mathbf{a}_t) \right|^2}{(\mathbf{f}_m \otimes \tilde{\mathbf{w}}_m)^H \mathbf{R} (\mathbf{f}_m \otimes \tilde{\mathbf{w}}_m)}. \quad (3.111)$$

This equation can be further simplified by substituting the FTS weight vector  $\tilde{\mathbf{w}}_m$  from Eqn. (3.104) into Eqn. (3.111),

$$\text{SINR}_m = \frac{\sigma^2 \xi_t \left| \left( \mathbf{f}_m \otimes \tilde{\mathbf{R}}_m^{-1} \tilde{\mathbf{g}}_t \right)^H (\mathbf{b}_t(\bar{\omega}) \otimes \mathbf{a}_t) \right|^2}{\left( \mathbf{f}_m \otimes \tilde{\mathbf{R}}_m^{-1} \tilde{\mathbf{g}}_t \right)^H \mathbf{R} \left( \mathbf{f}_m \otimes \tilde{\mathbf{R}}_m^{-1} \tilde{\mathbf{g}}_t \right)}, \quad (3.112)$$

where  $\tilde{\mathbf{g}}_t$  is the tapered spatial steering vector,

$$\tilde{\mathbf{g}}_t = \mathbf{t}_a \odot \mathbf{a}_t. \quad (3.113)$$

Figure 3.20 compares the SINR Loss  $L_{\text{SINR}}$  for the FTS algorithm using several different Doppler filter bank tapers,  $\mathbf{t}_b$ . No taper has been used in the spatial domain. Since there are 18 pulses in the CPI, the Doppler filter contains 18 bins. There is an SINR expression for each bin. The figures plot the envelope of all the SINR expressions for each bin. As the figure illustrates, the lower sidelobes in the Doppler filter bank increase the SINR. This trend is expected since the primary benefit of the lower sidelobes is suppression of the interference.

Figure 3.20 also brings forth an unexpected point. As the quality of the Doppler filter increases (as measured by the lower sidelobes due to the window function used), the performance of the FTS algorithm approaches the optimum for fully adaptive STAP. The non-adaptive Doppler filter is performing almost as well as the adaptive Doppler filter. The trade-off occurs on the edges of the normalized Doppler spectrum. Considerable roll-off occurs above  $\bar{\omega} = 0.8$  and below  $\bar{\omega} = 0.2$  for the Blackman Harris Doppler window.

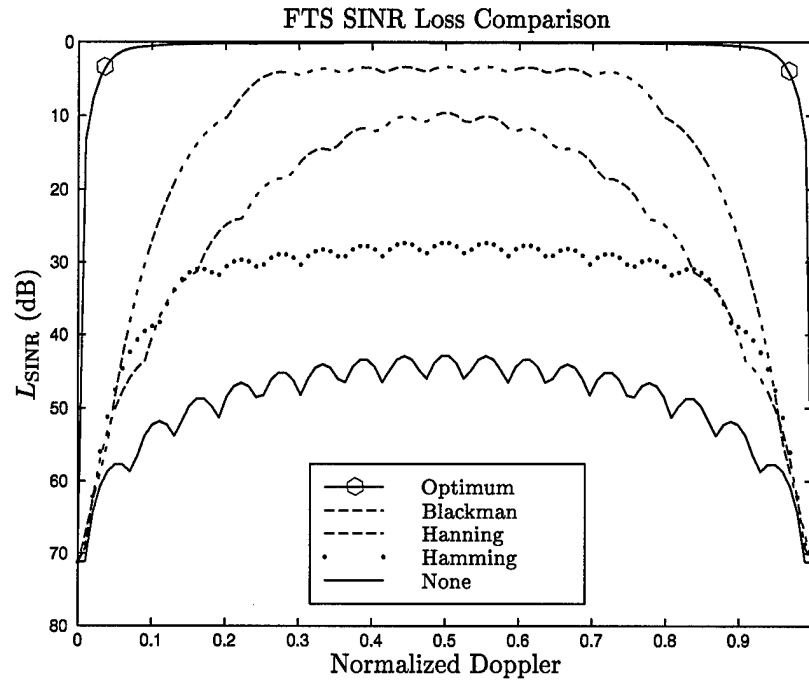


Figure 3.20 Comparison of  $L_{\text{SINR}}$  for the FTS algorithm with four Doppler Filter bank windows and no spatial taper. The optimum SINR curve corresponds to a fully adaptive algorithm ( $18 \times 18 = 324$  DOF) with no tapers. The FTS algorithm is adaptive in the spatial domain only, resulting in 18 DOF for this example. The Blackman Harris window results in the greatest SINR for the FTS algorithm because of the -92 dB sidelobes. The Hamming and Hanning window sidelobes are -43 dB and -35 dB, respectively. The greater sidelobe roll-off of the Hanning window (-18 dB/Octave) allows it to outperform the Hamming window (-6 dB/Octave). Window parameters taken from Harris [11].

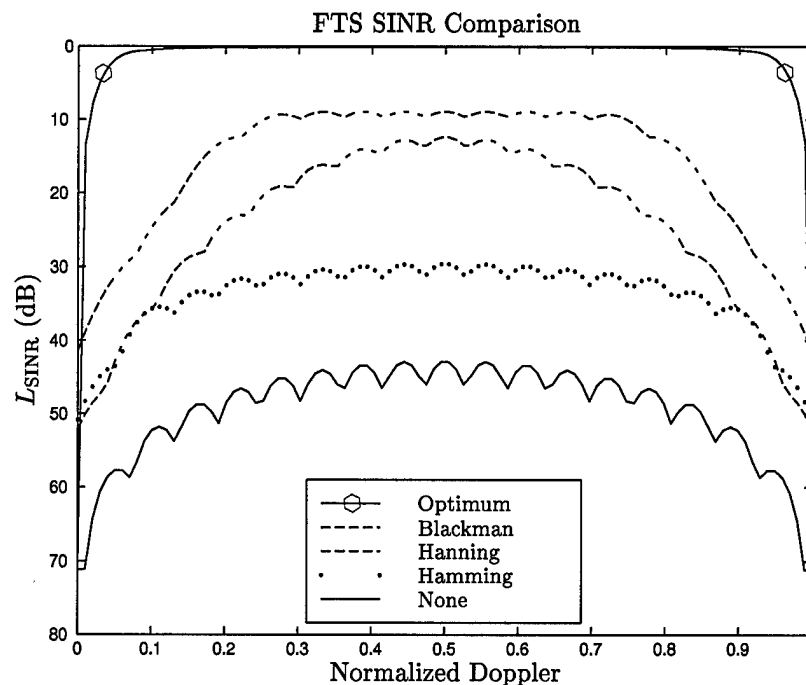


Figure 3.21 Comparison of  $L_{\text{SINR}}$  for the FTS algorithm with four Doppler Filter bank windows and the same spatial window as the Doppler window. The introduction of the spatial taper reduced the SINR as expected. The optimum SINR curve corresponds to a fully adaptive algorithm ( $18 \times 18 = 324$  DOF) with no tapers. The FTS algorithm is adaptive in the spatial domain only, resulting in 18 DOF for this example. The Blackman Harris window used both in the non-adaptive Doppler filter and the spatial taper results in the greatest SINR for the FTS algorithm because of the -92 dB sidelobes. The Hamming and Hanning window sidelobes are -43 dB and -35 dB, respectively. The greater sidelobe roll-off of the Hanning window (-18 dB/Octave) allows it to outperform the Hamming window (-6 dB/Octave). Window parameters taken from Harris [11].

As a final comparison, a spatial filter matching the Doppler filter was added to the FTS algorithm. These SINR Loss plots are shown in Fig. 3.21. Each curve shows a decrease in SINR due to the addition of the spatial taper.

### 3.6 Summary

This chapter introduced the basic fundamentals of fully adaptive STAP. The fully adaptive algorithm discussed placed a null at zero degrees azimuth to suppress mainbeam clutter. It also placed nulls on any barrage noise jammers present. However, one point not addressed is how the

radar informed the operator of the position of the antenna nulls. If the algorithm did a very good job of filtering out the noise jammers, it seems theoretically possible for the radar operator to receive no indication of any targets coming in at that azimuth/elevation (they would be filtered out). An obvious ECCM tactic would be to equip an aircraft with a barrage noise jammer. A radar equipped with this algorithm would never know the target was there.

Other items of interest were the series of nulls in the antenna pattern. These graphs not only showed the deep antenna nulls at jammer azimuths and zero Doppler (mainbeam clutter) but also a series of nulls (13 of them) across the azimuth range. These nulls range from -50 to -70 dB compared to the -90 of the jammer nulls. These nulls not only filter out undesired returns but also any desired returns at these azimuths. The Doppler pattern suffers the same problem.

Because of the computational issues associated with the fully adaptive STAP algorithm, the partially adaptive FTS algorithm was introduced. There are many other reduced dimension algorithms available, this one was introduced because of its simplicity and value in explaining reduced dimension concepts.

## *IV. Reduced Dimensionality Through the Cross Spectral Metric Algorithm*

### *4.1 Introduction*

As mentioned previously, Ward's [21] work serves as an excellent tutorial for STAP. Chapter III is a summary of this work, serving to introduce the basics of STAP and develop an understanding of the important issues. This next chapter delves deeper into the knowledge required to meet the main objective of the thesis: determination of the secondary data support required for one of the recent approaches to reduced dimension STAP. A secondary objective involves determining the effect of non-homogeneities in the secondary data of this model. This objective also requires the information found here.

The Cross Spectral Metric (CSM) algorithm just recently introduced by Goldstein [6] was chosen as the algorithm of study. There are several reasons for this choice. The foremost reason is its deviation from the standard Sample Matrix Inversion (SMI) approach found in Chapter III. By deviating from this approach, the algorithm brings into question the applicability of work done by Reed [15] on required secondary data support.

The introduction of the CSM algorithm begins with a brief exploration of some of the fundamental papers in the field of reduced dimension STAP. From here it progresses into formulating the direct form STAP problem as described in Chapter III into a generalized sidelobe canceler model. This model allows an easy development of the CSM algorithm.

### *4.2 Fundamentals*

In light of these goals, let's begin by mentioning one of the fundamental papers in this field. This article [2] developed the expression introduced in Chapter III for the optimum antenna weights in the adaptive processor. As shown by Brennan and Reed, the probability of false alarm  $P_{FA}$  and

probability of detection  $P_D$  for a signal model given by  $\mathbf{x} = \mathbf{v} + \mathbf{n}$  are

$$\begin{aligned} P_{FA} &= e^{-\frac{\beta^2}{2}} \\ P_D &= Q\left(\frac{|\mathbf{w}^T \mathbf{v}|}{\sqrt{\mathbf{w}^H \mathbf{R} \mathbf{w}}}, \beta\right), \end{aligned} \quad (4.1)$$

where  $\mathbf{w}$  represents the complex valued weight vector for the adaptive processor,  $\mathbf{v}$  the desired signal vector or space-time steering vector,  $\mathbf{n}$  the noise vector composed of zero mean samples with covariance matrix  $\mathbf{R}$ , and  $\beta$  is a normalized threshold. The  $Q$  function is defined as

$$Q(\alpha, \beta) = \int_{\beta}^{\infty} \nu e^{-\frac{\nu^2 + \alpha^2}{2}} I_0(\alpha \nu) d\nu, \quad (4.2)$$

where  $I_0(x)$  is the modified zero-order Bessel function of the first kind. These expressions were developed by first deriving the likelihood ratio test for the detection of a target in the presence of noise and then applying the optimum Bayes test. This is the origin of the parameter  $\beta$ . Since the parameter is not of particular concern to this discussion, it is only mentioned in passing here.

Brennan and Reed then proceeded to maximize the above expression for  $P_D$ . They first proved the function given is a monotonically increasing function of  $\alpha = \frac{|\mathbf{w}^T \mathbf{v}|}{\sqrt{\mathbf{w}^H \mathbf{R} \mathbf{w}}}$ , thereby showing the probability of detection will be highest for the filter  $\mathbf{w}$  that maximizes  $\alpha$ . The second parameter,  $\beta$ , is of no concern because it is the normalized threshold from the optimum Bayes test and will remain constant.

Since they had already proven  $P_D(\alpha)$  is monotonically increasing for  $\alpha$ , it follows the function is also increasing for  $\alpha^2$ . Squaring  $\alpha$  shows it is the Signal-to-Noise-Ratio (SNR),

$$\alpha^2 = \frac{|\mathbf{w}^T \mathbf{v}|^2}{\mathbf{w}^H \mathbf{R} \mathbf{w}}. \quad (4.3)$$



The problem of maximizing the probability of detection has now been shown to be equivalent to maximizing the SNR, an important insight. Because the noise covariance matrix is Hermitian ( $\mathbf{R} = \mathbf{R}^H$ ), then it can be diagonalized through the use of a unitary matrix. This is the eigenvalue decomposition of  $\mathbf{R}$ . Using this decomposition, Brennan and Reed showed the maximum SNR is obtained by letting

$$\mathbf{w} = k\mathbf{R}^{-1}\mathbf{v}^*, \quad (4.4)$$

where  $k$  is a complex valued constant. This is the unconstrained maximization of the SNR. The maximum SNR is easily found through substituting (4.4) into (4.3),

$$\max_{\mathbf{w}} \left( \frac{S}{N} \right) = \mathbf{v}^H \mathbf{R}^{-1} \mathbf{v}. \quad (4.5)$$

Previously (see Chapter III) the optimum adaptive processor weights were defined as  $\mathbf{R}^{-1}\mathbf{v}$ , where the conjugation operation on the steering vector  $\mathbf{v}$  seems to be missing. The difference arises because Brennan and Reed define the output of the adaptive processing filter as  $y = \mathbf{w}^T \boldsymbol{\chi}$  instead of  $y = \mathbf{w}^H \boldsymbol{\chi}$ . Carrying this difference through their equations results in the same expression.

Equation (4.4) is the basis for full dimension STAP. Because of the large dimensionality associated with the interference covariance matrix  $\mathbf{R}$ , the computational complexity of inverting it is still too cumbersome for real time operation. Furthermore, the fluid airborne environment necessitates estimating  $\mathbf{R}$  from data obtained within a Coherent Processing Interval (CPI). The estimate itself is difficult to obtain because the amount of secondary data needed when the number of Degrees Of Freedom (DOF) retains full dimensionality.

Considering these practical road blocks, Brennan and Reed discuss alternative methods of implementing the optimum adaptive filter. In particular a discussion of the method of steepest as-

cent is proposed. However, better methods have arisen since the paper was written. This discussion will now focus on one of those.

#### 4.3 *Translating Direct Form STAP into a Generalized Sidelobe Canceler Form*

Due to the computational burden associated with retaining full dimensionality and using all DOF within the adaptive processor, an optimum method is needed to generate a reduced dimension algorithm. The objective is to trade DOF for the ability to estimate the covariance matrix and calculate its inverse in real time, subject to both the physical and computational constraints associated with an airborne radar system.

The heart of a reduced dimension STAP algorithm is throwing away DOF. Determining which DOF are the most important for the current interference plus noise covariance matrix estimate is the problem at hand. For example, the Factored Time-Space (FTS) method simply throws away all Doppler DOF by non-adaptively filtering in the temporal domain. Absolutely no consideration is given to the impact this approach has on the ability to effectively null out the significant sources of interference. Poor selection can result in the adaptive processor expending all DOF to null out a large clutter source, with nothing left over to form a mainbeam in the direction of the target.

Goldstein and Reed have come up with an optimum way to select a reduced subset of the DOF. This subset maximizes the ability of the adaptive processor to null out large sources of interference, i.e. it maximizes the Signal-to-Interference plus Noise Ratio (SINR). First, begin by defining the normalized space-time steering vector  $\mathbf{s}$  as

$$\mathbf{s} = \frac{\mathbf{v}(\vartheta_t, \bar{\omega}_t)}{\sqrt{\mathbf{v}^H(\vartheta_t, \bar{\omega}_t)\mathbf{v}(\vartheta_t, \bar{\omega}_t)}}, \quad (4.6)$$

where  $\mathbf{v}$  was previously defined in Chapter III as the Kronecker product of the spatial and temporal steering vectors,  $\mathbf{a}(\vartheta_t)$  and  $\mathbf{b}(\bar{\omega}_t)$  respectively.

The purpose of any adaptive processing algorithm for a radar is to enhance target detection through maximizing the probability of detection  $P_D$  and minimizing the probability of false alarm  $P_{FA}$ . A more complete discussion of these two metrics and the associated hypothesis testing is found in Appendix C. Goldstein breaks the detection problem into two parts.

- First, the output SINR of the adaptive processor is maximized subject to the constraint of a normalized response in the direction of the steering vector.
- Second, the output of the adaptive processor is then subject to the hypothesis testing mechanisms mentioned above and discussed in detail in Appendix C.

As developed by Reed [2], the *unconstrained* optimization of the SINR was given in Eqn. (4.4), where the coefficient  $k$  was a complex valued constant. The problem with this solution is there is incidental cancellation of the desired signal returns. Applebaum then developed a method to calculate the optimum adaptive processor weights subject to the constraint of a normalized response in the direction of the steering vector. These adaptive weights are given by [1]

$$\mathbf{w}_{\text{SINR}} = \frac{\mathbf{R}^{-1}\mathbf{s}}{\mathbf{s}^H\mathbf{R}^{-1}\mathbf{s}}, \quad (4.7)$$

where the constraint is  $\mathbf{w}_{\text{SINR}}^H\mathbf{s} = 1$ . Substituting the above expression into the constraint shows it is satisfied. The output  $y$  of the adaptive processor is then

$$y = \mathbf{w}_{\text{SINR}}^H\boldsymbol{\chi} = \frac{\mathbf{s}^H\mathbf{R}^{-1}\boldsymbol{\chi}}{\mathbf{s}^H\mathbf{R}^{-1}\mathbf{s}}. \quad (4.8)$$

Next, direct form STAP is transformed into a Generalized Sidelobe Canceler (GSC) form by Goldstein. Consider the use of a unitary  $NM \times NM$  transformation matrix  $\mathbf{T}$ , where  $N$  and  $M$  are the spatial and temporal degrees of freedom. The resulting output SINR after this transformation is identical to the direct form processing architecture, where  $y = \mathbf{w}_{\text{SINR}}^H\boldsymbol{\chi}$ . Figure 4.1 shows a block diagram of the direct form model.

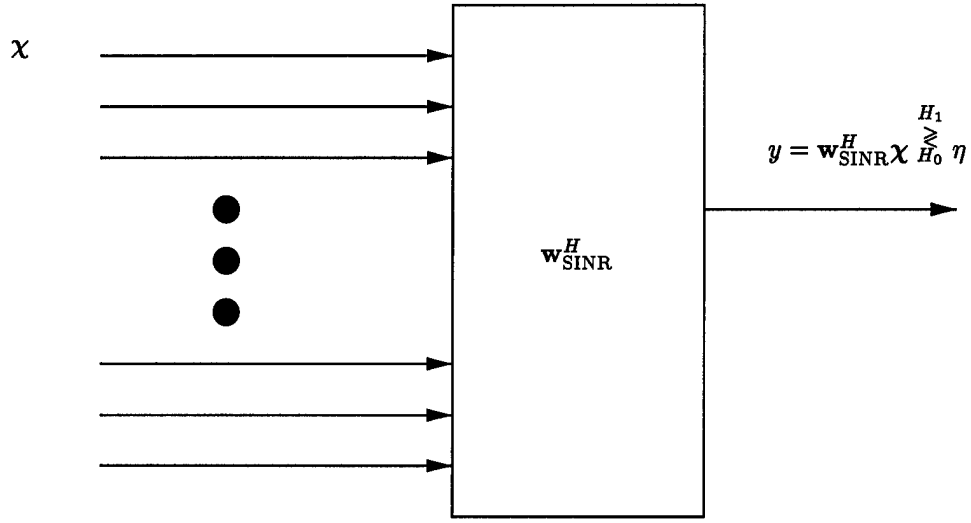


Figure 4.1 The direct form adaptive processor block diagram showing the optimum constrained filter weights.

The form of the unitary transformation matrix  $\mathbf{T}$  is

$$\mathbf{T} = \begin{bmatrix} \mathbf{s}^H \\ \mathbf{B} \end{bmatrix}, \quad (4.9)$$

where  $\mathbf{B}$  represents a blocking matrix. The blocking matrix projects the data vector  $\chi$  onto the null space of the steering vector  $\mathbf{s}$ , i.e.  $\mathbf{B}\mathbf{s} = \mathbf{0}$ . Therefore,  $\mathbf{B}$  blocks the desired signal and allows all other signals to pass. The size of the blocking matrix is  $NM - 1 \times NM$ . Any full row rank matrix satisfying  $\mathbf{B}\mathbf{s} = \mathbf{0}$  is a valid signal blocking matrix [7]. See Appendix B for a discussion on determining the blocking matrix. Using these definitions, the transformed data vector is

$$\tilde{\chi} = \mathbf{T}\chi = \begin{bmatrix} \mathbf{s}^H \chi \\ \mathbf{B}\chi \end{bmatrix} = \begin{bmatrix} d \\ \mathbf{b} \end{bmatrix}, \quad (4.10)$$

where  $\mathbf{b}$  is denoted the noise-subspace data vector because it contains the projection of the data vector onto the noise subspace. A block diagram of this new model is shown in Fig. 4.2. Haimovich [10]

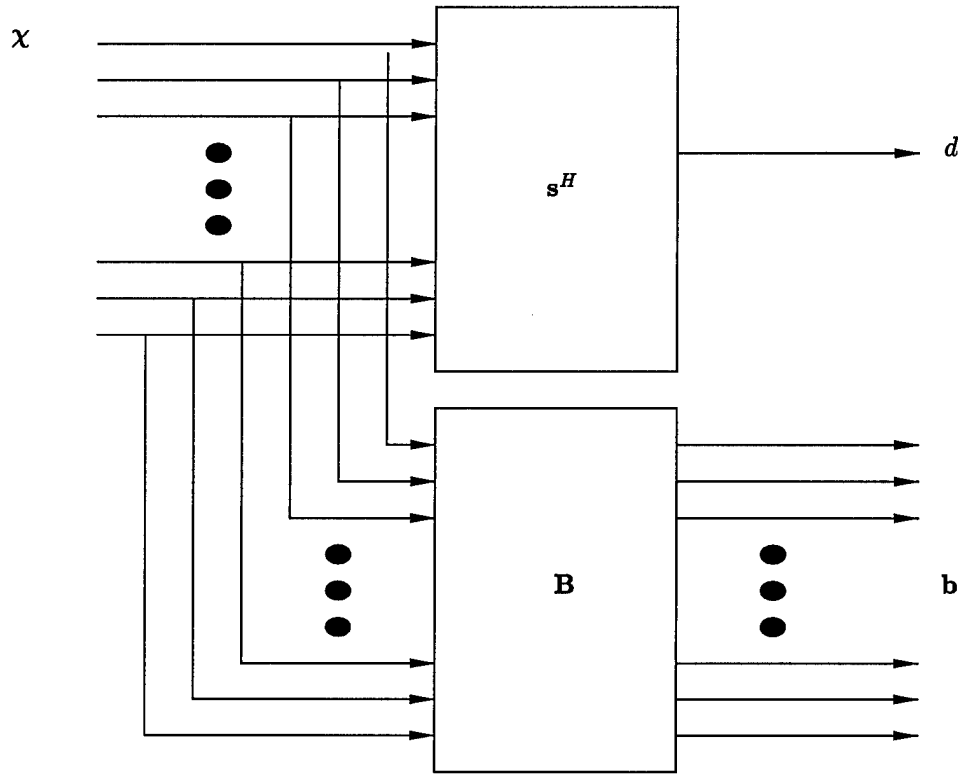


Figure 4.2 A block diagram of the blocking matrix concept used in the formulation of the GSC model of STAP.

discusses determination of the signal blocking matrix for a uniformly spaced array and spatial steering vector.

The covariance matrix of the data is found through the following manipulations,

$$\mathbf{R}_{\tilde{\mathbf{x}}} = \mathcal{E} \{ \tilde{\mathbf{x}} \tilde{\mathbf{x}}^H \} = \mathcal{E} \{ \mathbf{T} \mathbf{x} \mathbf{x}^H \mathbf{T}^H \} = \mathbf{T} \mathcal{E} \{ \mathbf{x} \mathbf{x}^H \} \mathbf{T}^H = \mathbf{T} \mathbf{R} \mathbf{T}^H. \quad (4.11)$$

Then substituting Eqn. (4.9) into Eqn. (4.11) results in

$$\mathbf{R}_{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{s}^H \\ \mathbf{B} \end{bmatrix} \mathbf{R} \begin{bmatrix} \mathbf{s} & \mathbf{B}^H \end{bmatrix} = \begin{bmatrix} \mathbf{s}^H \mathbf{R} \mathbf{s} & \mathbf{s}^H \mathbf{R} \mathbf{B}^H \\ \mathbf{B} \mathbf{R} \mathbf{s} & \mathbf{B} \mathbf{R} \mathbf{B}^H \end{bmatrix} = \begin{bmatrix} \sigma_d^2 & \mathbf{r}_{\mathbf{b}d}^H \\ \mathbf{r}_{\mathbf{b}d} & \mathbf{R}_{\mathbf{b}} \end{bmatrix}, \quad (4.12)$$

where this partitioned matrix's parameters are implicitly defined when transitioning from the next to last step to the result, i.e.

$$\begin{aligned}\sigma_d^2 &= \mathbf{s}^H \mathbf{R} \mathbf{s}, \\ \mathbf{r}_{bd} &= \mathcal{E} \{ \mathbf{b} d^* \} = \mathbf{B} \mathbf{R} \mathbf{s}, \\ \mathbf{R}_b &= \mathcal{E} \{ \mathbf{b} \mathbf{b}^H \} = \mathbf{B} \mathbf{R} \mathbf{B}^H.\end{aligned}\tag{4.13}$$

Now let the transformation matrix  $\mathbf{T}$  operate on the steering vector. Since the steering vector was normalized previously a unit vector results,

$$\mathbf{T} \mathbf{s} = \begin{bmatrix} \mathbf{s}^H \\ \mathbf{B} \end{bmatrix} \mathbf{s} = \begin{bmatrix} \mathbf{s}^H \mathbf{s} \\ \mathbf{B} \mathbf{s} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{e}_1.\tag{4.14}$$

Given these definitions, the optimal weight vector for the GSC adaptive processing model  $\mathbf{w}_{\text{GSC}}$  becomes

$$\mathbf{w}_{\text{GSC}} = \frac{\mathbf{R}_{\tilde{\chi}}^{-1} \tilde{\mathbf{s}}}{\tilde{\mathbf{s}}^H \mathbf{R}_{\tilde{\chi}}^{-1} \tilde{\mathbf{s}}} = \frac{\mathbf{R}_{\tilde{\chi}}^{-1} \mathbf{e}_1}{\mathbf{e}_1^H \mathbf{R}_{\tilde{\chi}}^{-1} \mathbf{e}_1} = \begin{bmatrix} 1 \\ -\mathbf{w}_b \end{bmatrix},\tag{4.15}$$

where  $\tilde{\mathbf{s}}$  is the transformed steering vector and the weight vector  $\mathbf{w}_b$  is to be defined shortly. This form of the problem is best viewed in a block diagram format, where the reasoning for the 1 at the beginning of the weight vector  $\mathbf{w}_{\text{GSC}}$  becomes readily apparent. Figure 4.3 shows the block diagram.

As the figure visually illustrates, the incoming signal is projected onto the steering vector. This projection passes the component of  $\chi$  that contains the target *and* any noise or interference

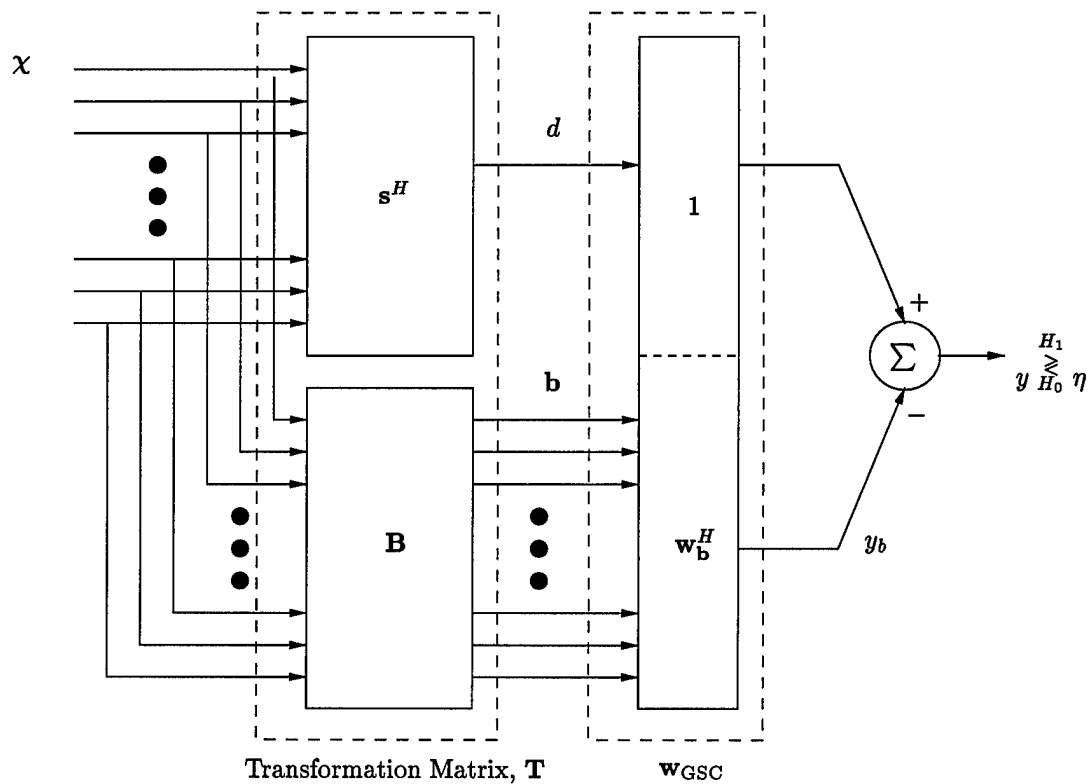


Figure 4.3 The full dimension GSC block diagram showing the transformation matrix  $\mathbf{T}$  and the GSC optimum weight vector  $\mathbf{w}_{\text{GSC}}$ .

that projects onto  $\mathbf{s}$ . The output of this operation is the scalar containing the desired component,  $d$ . Assuming the incoming data is the desired steering vector, i.e. there is a target in the direction the radar is looking, then  $\mathbf{x}$  becomes  $\mathbf{s}$  and  $d$  is equal to  $\mathbf{s}^H \mathbf{s}$ . This is the output of a filter with complex weights  $\mathbf{s}$ . These weights give a maximum response in the direction of the steering vector  $\mathbf{s}$ . Hence, the output is an estimate of the desired signal.

The operation of the GSC model can be viewed in two steps. The upper branch estimates the incoming signal at the direction of interest. Since there is also noise and interference coming from this direction (and all directions), the target estimate is corrupted by the noise and interference. The lower branch of the GSC attempts to estimate the noise and interference from the statistics of the remaining data not oriented in the direction of the steering vector. The incoming data vector is projected onto the subspace of the blocking matrix  $\mathbf{B}$ . This blocks any components of the incoming data vector matching the steering vector since  $\mathbf{B}\mathbf{s} = \mathbf{0}$ . The weight vector  $\mathbf{w}_b$  is then applied to  $\mathbf{b}$ , a vector containing only the noise and interference. These weights basically generate an approximation to the interference contained in  $d$  based on the interference in  $\mathbf{b}$ . This interference estimate is then subtracted from the signal plus interference and noise in the upper branch resulting in an ideally pure target/signal estimate corrupted only by noise applied to the radar's threshold detection.

The optimum value of the weight vector  $\mathbf{w}_b$  shown in Fig. 4.3 and Eqn. (4.15) is determined by the Wiener solution [8],

$$\mathbf{w}_b = \mathbf{R}_b^{-1} \mathbf{r}_{bd}, \quad (4.16)$$

where the cross correlation vector and autocorrelation matrix were defined previously.

Figure 4.3 is the model of fully adaptive STAP using the GSC. Just as before, the computational costs drive the need for a reduced dimension application. Some of the features of this



model will be used to develop a reduced dimension algorithm producing a maximum SINR with the available DOF.

Since the purpose of the lower branch of the GSC is to gain an estimate of the interference contained in the upper branch, the development now assumes  $\chi$  is interference only. This allows the analysis to parallel that of a Wiener filter, the upper branch being the desired component  $d$  (interference only) and the lower branch the interference estimate  $\hat{d}$ . Using this interpretation, the output of the GSC is the error between the interference estimate and the true interference value. However, it is important not to lose sight of the fact that the target signal is generally included in  $d$  also.

Care must be taken to distinguish interference from noise in this development. Interference is meant to include all correlated undesired components while the term noise refers to all uncorrelated undesired components. This distinction is made because only the correlated components can be filtered. This limitation is a direct result of the filter being based on the statistics, or correlation, of the secondary data. For the interference estimate to accurately reflect the true interference in the desired signal, the interference samples in the secondary data must be correlated to the interference found in the desired signal.

Now that a suitable model is available, let's analyze it's use in a reduced dimension application. From this analysis, we can then choose a suitable metric for selecting which DOF maximize the output SINR with reduced dimensionality. First, express the covariance matrix  $\mathbf{R}_b$  for the data vector  $\mathbf{b}$  in terms of its eigenvalue/eigenvector decomposition. This decomposition is given by

$$\mathbf{R}_b = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H, \quad (4.17)$$

where  $\mathbf{U}$  is the unitary matrix containing the eigenvectors and  $\mathbf{\Lambda}$  is the diagonal matrix containing the eigenvalues. In the derivation of the following equations, the unitary property allows the use of  $\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}$ . The size of  $\mathbf{U}$  and  $\mathbf{\Lambda}$  is  $(NM - 1) \times (NM - 1)$ . Now, if we project the vector

$\mathbf{b}$  to a different coordinate space using the unitary matrix of  $\mathbf{U}^H$ ,

$$\mathbf{p} = \mathbf{U}^H \mathbf{b}, \quad (4.18)$$

then we can start viewing the matrix  $\mathbf{U}$  in terms of reducing the dimension of the problem. Let's first examine the statistics of  $\mathbf{p}$ , the vector Goldstein [6] refers to as a principal coordinate process. The covariance matrix  $\mathbf{R}_p$  of  $\mathbf{p}$  is given by

$$\mathbf{R}_p = \mathcal{E} \{ \mathbf{p} \mathbf{p}^H \} = \mathbf{U}^H \mathbf{R}_b \mathbf{U} = \mathbf{\Lambda}. \quad (4.19)$$

The cross correlation between  $\mathbf{p}$  and  $d$  is

$$\mathbf{r}_{pd} = \mathcal{E} \{ \mathbf{p} d^* \} = \mathbf{U}^H \mathbf{r}_{bd}. \quad (4.20)$$

And finally, the Wiener-Hopf solution for the optimum weight vector  $\mathbf{w}_p$  in the full dimension GSC with the eigenvector matrix projection is

$$\mathbf{w}_p = \mathbf{R}_p^{-1} \mathbf{r}_{pd} = \mathbf{U}^H \mathbf{w}_b. \quad (4.21)$$

A block diagram of the full dimension GSC model using the full eigenvector matrix as a projection is shown in Fig. 4.4. The output of the GSC model can be inferred just from the diagram alone or by going back through the equations. This output is

$$y = (\mathbf{s}^H - \mathbf{w}_b^H \mathbf{B}) \chi. \quad (4.22)$$

At this point in Goldstein's derivation, the GSC model is still using the full dimensionality of the problem. DOF reduction is performed through the use of the unitary transformation matrix

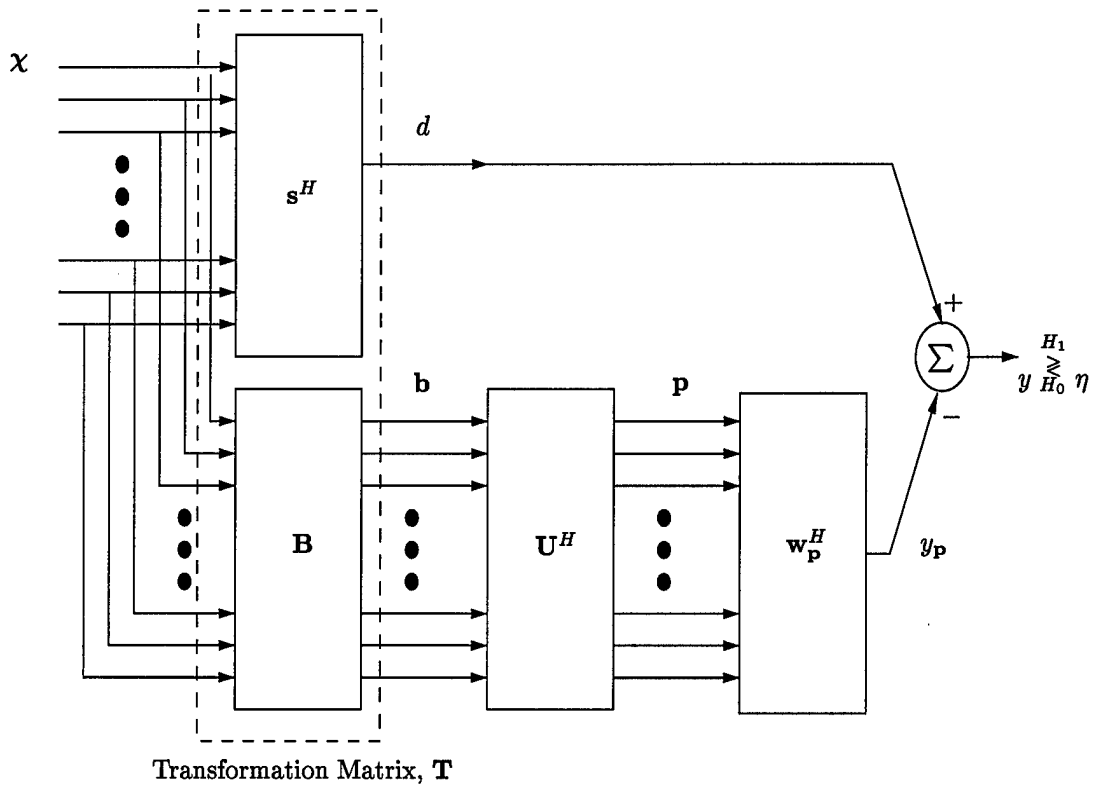


Figure 4.4 The full dimension GSC block diagram with eigenvector projection.

of eigenvalues  $\mathbf{U}$ . In the case of reduced dimensionality, only a subset of the columns of  $\mathbf{U}$  are used. This subset of the total number of eigenvectors gives the new transformation  $\mathbf{u}$  a size of  $(NM - 1) \times O$ , where  $\mathbf{u}$  is composed of  $O$  columns of  $\mathbf{U}$ . Multiplying  $\mathbf{b}$  by  $\mathbf{u}$  reduces the dimension to  $O$ . A block diagram of the reduced dimension GSC model is shown in Fig. 4.5. As inferred by the figure, the output of the dimension reduction  $\mathbf{z}$  is defined as

$$\mathbf{z} = \mathbf{u}^H \mathbf{b}. \quad (4.23)$$

The next problem is determining which eigenvectors represent the best choice for the construction of  $\mathbf{u}$ . Goldstein begins this process by looking at the reduced dimension covariance matrix

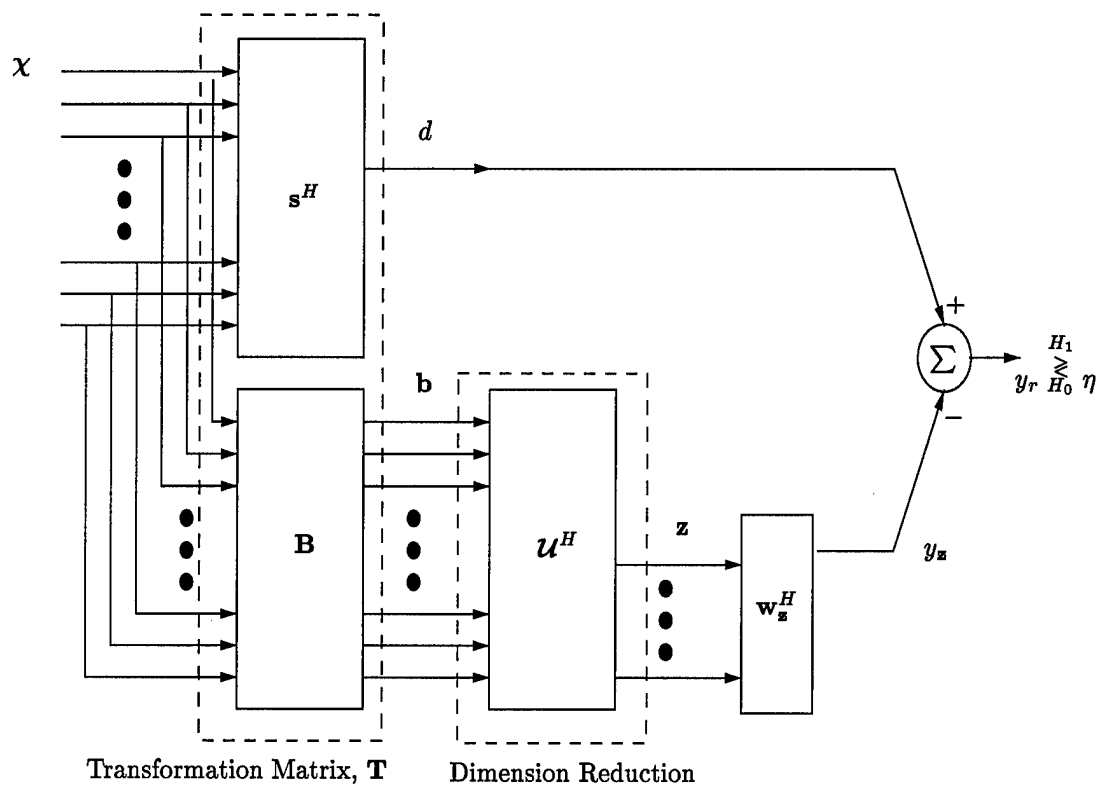


Figure 4.5 The reduced dimension GSC block diagram using  $O$  eigenvectors as a projection.

$\mathbf{R}_z$ ,

$$\mathbf{R}_z = \mathbf{U}^H \mathbf{R}_b \mathbf{U} = \mathbf{\Lambda}_O, \quad (4.24)$$

where  $\mathbf{\Lambda}_O$  is the diagonal matrix composed of the eigenvalues corresponding to the eigenvectors used to construct  $\mathbf{U}$ . The cross correlation vector  $\mathbf{r}_{zd}$  between the reduced dimension noise subspace vector  $\mathbf{z}$  and  $d$  is

$$\mathbf{r}_{zd} = \mathcal{E} \{ \mathbf{z} d^* \} = \mathbf{U}^H \mathbf{r}_{bd}. \quad (4.25)$$

Using these statistics, the Wiener solution  $\mathbf{w}_z$  for the  $O$  dimensional weight vector is

$$\mathbf{w}_z = \mathbf{R}_z^{-1} \mathbf{r}_{zd} = \mathbf{\Lambda}_O^{-1} \mathbf{U}^H \mathbf{r}_{bd}. \quad (4.26)$$

All of the structures necessary to build the adaptive filter are in place. The only issue in need of consideration is the choice of eigenvectors used to compose the dimension reducing matrix  $\mathbf{U}$ . Goldstein states [8] the most popular technique for this subspace selection is based on the principal components method.

The principal components method chooses the set of eigenvectors resulting in the closest approximation to the full covariance matrix  $\mathbf{R}_b$ . This is the set of eigenvectors corresponding to the  $O$  largest eigenvalues of the matrix. The problem with this method is that it results in the optimum representation of the covariance matrix. This is not the point of the problem.

The eigenvectors should be chosen to maximize the SINR of the adaptive filter. As shown by Goldstein, this selection is not necessarily the same as choosing the eigenvectors corresponding to the largest eigenvalues. This selection is more than just a function of the covariance matrix, it is also a function of the cross correlation between the beamformed signal  $d$  and the noise process  $\mathbf{b}$ .

Goldstein provides the derivation of the optimum selection of the eigenvectors in another paper [6]. The eigenvectors are selected to satisfy a Minimum Mean Square Error (MMSE) criterion. The output of the full dimension GSC model was previously shown in Eqn. (4.22) to be

$$y = (\mathbf{s}^H - \mathbf{w}_b^H \mathbf{B}) \chi \quad (4.27)$$

and the Wiener-Hopf solution was shown in Eqn. (4.16) as

$$\mathbf{w}_b = \mathbf{R}_b^{-1} \mathbf{r}_{bd}. \quad (4.28)$$

Using these two equations, the MMSE  $P$  is

$$P = \mathcal{E} \{ |y|^2 \} = \sigma_d^2 - \mathbf{r}_{bd}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd}, \quad (4.29)$$

where the cross correlation vector and covariance matrix of the transformed data vector were defined previously. The MMSE is conserved by a unitary transformation [6], therefore the MMSE associated with the full dimension GSC model shown in Fig. 4.4 remains the same.

Goldstein uses the term MMSE because we are considering the case of noise interference only. Using this assumption, the output of the GSC model should be zero given a perfect estimate of the interference. Hence, at this point in the derivation any output of the GSC is an error in the interference estimate.

Denote the output of the reduced dimension GSC model as  $y_r$ , following Fig. 4.5. The output in terms of the transformed data vector  $\mathbf{b}$  is

$$y_r = d - y_z = d - \mathbf{w}_z^H \mathbf{U}^H \mathbf{b}. \quad (4.30)$$

Adding and subtracting the output of the full dimension GSC does not modify the equation,

$$y_r = y - y + d - y_z. \quad (4.31)$$

Referencing Fig. 4.4 shows we can write  $y$  as  $d - y_p$ . Therefore, substituting for *one* of the terms gives

$$y_r = y - d + y_p + d - y_z = y + y_p - y_z, \quad (4.32)$$

or in terms of the matrices already defined,

$$y_r = \begin{bmatrix} 1 & \mathbf{w}_p^H \mathbf{U}^H - \mathbf{w}_z^H \mathcal{U}^H \end{bmatrix} \begin{bmatrix} y \\ \mathbf{b} \end{bmatrix}. \quad (4.33)$$

Using this expression, the  $(NM - 1) \times 1$  vector  $\mathbf{e}$  representing the error between the reduced dimension weight vector and the full dimension weight vector is defined by Goldstein as

$$\mathbf{e} = \mathbf{U} \mathbf{w}_p - \mathcal{U} \mathbf{w}_z. \quad (4.34)$$

With these forms, the mean-square value of the reduced dimension GSC model output is

$$\mathcal{E} \{ |y_r|^2 \} = \mathcal{E} \{ |y + \mathbf{e}^H \mathbf{b}|^2 \}. \quad (4.35)$$

Expanding the quadratic uncovers the error found in the full dimension GSC model plus terms due to the reduced dimensionality,

$$\begin{aligned}
\mathcal{E} \{ |y + \mathbf{e}^H \mathbf{b}|^2 \} &= \mathcal{E} \{ (y + \mathbf{e}^H \mathbf{b})(y + \mathbf{e}^H \mathbf{b})^H \} \\
&= \mathcal{E} \{ |y|^2 + y \mathbf{b}^H \mathbf{e} + y^* \mathbf{e}^H \mathbf{b} + \mathbf{e}^H \mathbf{b} \mathbf{b}^H \mathbf{e} \} \\
&= P + \mathbf{e}^H \mathbf{R}_b \mathbf{e} + \mathcal{E} \{ y \mathbf{b}^H \mathbf{e} \} + \mathcal{E} \{ y^* \mathbf{e}^H \mathbf{b} \}.
\end{aligned} \tag{4.36}$$

Ridding ourselves of the last two terms is interesting. Using Fig. 4.4 as a reference, examine the purpose of the model. The only input under consideration at this point is that of the interference. Therefore, the lower branch is an estimate of the interference in the upper branch and the output  $y$  is simply the difference between the estimate and the true value,

$$y = d - \hat{d}. \tag{4.37}$$

Again using the figure, the estimator is based on the noise/interference projection vector  $\mathbf{b}$  such that  $\hat{d} = \mathbf{w}_p^H \mathbf{U}^H \mathbf{b}$ . The full dimension Wiener filter was already determined in Eqn. (4.21) as

$$\mathbf{w}_p = \mathbf{R}_p^{-1} \mathbf{r}_{pd} = \mathbf{U}^H \mathbf{w}_b. \tag{4.38}$$

Therefore, the noise estimator becomes

$$\hat{d} = \mathbf{w}_b^H \mathbf{U} \mathbf{U}^H \mathbf{b} = \mathbf{w}_b^H \mathbf{b}, \tag{4.39}$$



where the property  $\mathbf{U}\mathbf{U}^H = \mathbf{I}$  is invoked for the unitary matrices. Using the Wiener solution for the above filter found in Eqn. (4.16), the estimator can be rewritten as

$$\hat{d} = \mathbf{r}_{\mathbf{b}d}^H \mathbf{R}_{\mathbf{b}}^{-1} \mathbf{b}, \quad (4.40)$$

where the Hermitian property of the covariance matrix has been invoked, i.e.  $\mathbf{R}_{\mathbf{b}}^H = \mathbf{R}_{\mathbf{b}}$ .

Now let's examine the third term of Eqn. (4.36) using the expression for the estimate of  $d$ ,

$$\begin{aligned} \mathcal{E} \{y \mathbf{b}^H \mathbf{e}\} &= \mathcal{E} \{ (d - \hat{d}) \mathbf{b}^H \} \mathbf{e} \\ &= \mathcal{E} \{ (d - \mathbf{r}_{\mathbf{b}d}^H \mathbf{R}_{\mathbf{b}}^{-1} \mathbf{b}) \mathbf{b}^H \} \mathbf{e} \\ &= \mathcal{E} \{ d \mathbf{b}^H \} \mathbf{e} - \mathcal{E} \{ \mathbf{r}_{\mathbf{b}d}^H \mathbf{R}_{\mathbf{b}}^{-1} \mathbf{b} \mathbf{b}^H \} \mathbf{e} \\ &= \mathbf{r}_{d\mathbf{b}} \mathbf{e} - \mathbf{r}_{\mathbf{b}d}^H \mathbf{R}_{\mathbf{b}}^{-1} \mathbf{R}_{\mathbf{b}} \mathbf{e} \\ &= \mathbf{r}_{\mathbf{b}d}^H \mathbf{e} - \mathbf{r}_{\mathbf{b}d}^H \mathbf{e} \\ &= 0, \end{aligned} \quad (4.41)$$

where the final result uses the fact  $\mathbf{r}_{d\mathbf{b}} = \mathbf{r}_{\mathbf{b}d}^H$ . The fourth term in Eqn. (4.36) can be shown to be equal to the Hermitian of the term just shown to equal zero,

$$\mathcal{E} \{y^* \mathbf{e}^H \mathbf{b}\} = \mathcal{E} \{y \mathbf{b}^H \mathbf{e}\}^H, \quad (4.42)$$

hence it is also zero.

It is interesting to note the above mathematics were not necessary to show the two terms are zero. Since the estimate of  $\hat{d}$  is the result of a Wiener filtering operation, then by definition the error is minimized. This minimization is another way of stating the error,  $y = d - \hat{d}$ , is orthogonal to the data vector  $\mathbf{b}$  used to generate the statistics for the Wiener filter. Since  $y$  at this point is

an expression of the error between  $d$  and  $\hat{d}$ , then  $y$  is orthogonal to  $\mathbf{b}$  and their inner product is obviously zero.

Either through the mathematical manipulation or the Wiener filtering rationalization, the expression in Eqn. (4.36) reduces to the one given by Goldstein [6]

$$\mathcal{E} \{ |y_r|^2 \} = \mathcal{E} \{ |y + \mathbf{e}^H \mathbf{b}|^2 \} = P + \mathbf{e}^H \mathbf{R}_b \mathbf{e}, \quad (4.43)$$

where  $P$  is the mean square output of the full dimension GSC model. This implies the error due to the dimension reduction is contained in the second term,  $\mathbf{e}^H \mathbf{R}_b \mathbf{e}$ . Minimization of the error is done via minimization of this term.

At this point in the derivation, we deviate from Goldstein's development to one of our own. Let us begin the minimization by first expressing the error term explicitly,

$$\mathbf{e}^H \mathbf{R}_b \mathbf{e} = \left( \mathbf{w}_p^H \mathbf{U}^H - \mathbf{w}_s^H \mathbf{U}^H \right) \mathbf{R}_b (\mathbf{U} \mathbf{w}_p - \mathbf{U} \mathbf{w}_s). \quad (4.44)$$

The covariance matrix  $\mathbf{R}_b$  was the subject of the previous eigenvalue/eigenvector decomposition. Using this decomposition, we can substitute into Eqn. (4.44),

$$\mathbf{e}^H \mathbf{R}_b \mathbf{e} = \left( \mathbf{w}_p^H \mathbf{U}^H - \mathbf{w}_s^H \mathbf{U}^H \right) \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H (\mathbf{U} \mathbf{w}_p - \mathbf{U} \mathbf{w}_s). \quad (4.45)$$

Distributing the unitary eigenvector matrices results in

$$\mathbf{e}^H \mathbf{R}_b \mathbf{e} = \left( \mathbf{w}_p^H \mathbf{U}^H \mathbf{U} - \mathbf{w}_s^H \mathbf{U}^H \mathbf{U} \right) \mathbf{\Lambda} (\mathbf{U}^H \mathbf{U} \mathbf{w}_p - \mathbf{U}^H \mathbf{U} \mathbf{w}_s) \quad (4.46)$$

and allows us to invoke the unitary property of the eigenvector matrix  $\mathbf{U}$  to further simplify the error expression to

$$\mathbf{e}^H \mathbf{R}_b \mathbf{e} = \left( \mathbf{w}_p^H - \mathbf{w}_z^H \mathcal{U}^H \mathbf{U} \right) \Lambda \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right). \quad (4.47)$$

If we pull the hermitian operation out so it operates on the entire left quantity and factor the diagonal eigenvalue matrix we see the equation is basically the magnitude squared of a simpler term,

$$\begin{aligned} \mathbf{e}^H \mathbf{R}_b \mathbf{e} &= \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right)^H \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right) \\ &= \left| \Lambda^{\frac{1}{2}} \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right) \right|^2, \end{aligned} \quad (4.48)$$

where the  $||$  operator denotes the magnitude of the vector. Because the covariance matrix  $\mathbf{R}$  is hermitian, then  $\mathbf{R}_b$  is also hermitian. For any hermitian matrix, the eigenvalues are real valued [3, page 119]. This allows us to take the hermitian of the diagonal eigenvalue matrix  $\Lambda^{\frac{1}{2}}$  without any effect, i.e.  $\left\{ \Lambda^{\frac{1}{2}} \right\}^H = \Lambda^{\frac{1}{2}}$ . Using Eqn. 4.48, minimization of the error term becomes

$$\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{\frac{1}{2}} \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right) \right|^2 \right\} = \min \left\{ \left| \Lambda^{\frac{1}{2}} \left( \mathbf{w}_p - \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right) \right| \right\}. \quad (4.49)$$

Now let's begin the analysis of this equivalent minimization term in the hopes of further simplifying it and developing an optimum method to choose the eigenvectors comprising the matrix  $\mathcal{U}$ . First, distribute the eigenvalue matrix across the quantity,

$$\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{\frac{1}{2}} \mathbf{w}_p - \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathcal{U} \mathbf{w}_z \right|^2 \right\}. \quad (4.50)$$

From Eqn. (4.21) we can substitute for  $\mathbf{w}_p$ ,

$$\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{w}_b - \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{u} \mathbf{w}_s \right|^2 \right\}. \quad (4.51)$$

Using Eqn. (4.16) to substitute for the  $NM - 1$  dimensioned weight vector  $\mathbf{w}_b$  and Eqn. (4.26) for the  $O$  dimensional weight vector  $\mathbf{w}_s$  results in

$$\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd} - \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\}. \quad (4.52)$$

Since we previously defined the eigenvector/eigenvalue decomposition of  $\mathbf{R}_b$  as  $\mathbf{U} \Lambda \mathbf{U}^H$ , then the inverse of  $\mathbf{R}_b$  is

$$\mathbf{R}_b^{-1} = \mathbf{U} \Lambda^{-1} \mathbf{U}^H, \quad (4.53)$$

allowing us to substitute into Eqn. (4.52) and again invoke the property of the unitary eigenvector matrix,

$$\begin{aligned} \min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) &= \min \left\{ \left| \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{U} \Lambda^{-1} \mathbf{U}^H \mathbf{r}_{bd} - \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\} \\ &= \min \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \Lambda^{\frac{1}{2}} \mathbf{U}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\}. \end{aligned} \quad (4.54)$$

The next step in the development involves the ordering of the eigenvectors and eigenvalues in the decomposition of  $\mathbf{R}_b$ . This decomposition can be expressed in the form of a summation,

$$\mathbf{R}_b = \mathbf{U} \Lambda \mathbf{U}^H = \sum_{i=0}^{NM-1} \lambda_i \nu_i \nu_i^H. \quad (4.55)$$

Examining this form shows the ordering of the columns of  $\mathbf{U}$  and  $\Lambda$  is totally insignificant as long as the eigenvalue/eigenvector relationship is maintained. At this point in the derivation, let us assume

the decomposition is ordered such that the first  $O$  columns of  $\mathbf{U}$  and  $\mathbf{\Lambda}$  contain the eigenvectors and eigenvalues corresponding to  $\mathbf{u}$ . Then these two matrices can be written in the form,

$$\mathbf{U} = \begin{bmatrix} \mathbf{u} & \mathbf{C} \end{bmatrix} \quad (4.56)$$

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_O & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{NM-1-O} \end{bmatrix}, \quad (4.57)$$

where  $\mathbf{C}$  is the matrix of eigenvectors not contained in  $\mathbf{u}$ ,  $\mathbf{\Lambda}_{NM-1-O}$  is the diagonal matrix of eigenvalues corresponding to  $\mathbf{C}$ , and  $\mathbf{0}$  represents a matrix of zeros.

Substituting this form of  $\mathbf{U}$  into Eqn. (4.54) allows us to write

$$\begin{aligned} \min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) &= \min \left\{ \left| \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{\Lambda}^{\frac{1}{2}} \begin{bmatrix} \mathbf{u}^H \\ \mathbf{C}^H \end{bmatrix} \mathbf{u} \mathbf{\Lambda}_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\} \\ &= \min \left\{ \left| \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{\Lambda}^{\frac{1}{2}} \begin{bmatrix} \mathbf{u}^H \mathbf{u} \\ \mathbf{C}^H \mathbf{u} \end{bmatrix} \mathbf{\Lambda}_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\} \\ &= \min \left\{ \left| \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{\Lambda}^{\frac{1}{2}} \begin{bmatrix} \mathbf{I}_O \\ \mathbf{0} \end{bmatrix} \mathbf{\Lambda}_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\}, \end{aligned} \quad (4.58)$$

where the final step takes advantage of the orthogonality of the eigenvectors. Furthermore, multiplying the matrix  $\begin{bmatrix} \mathbf{I}_O & \mathbf{0} \end{bmatrix}$  by  $\mathbf{\Lambda}^{\frac{1}{2}}$  simply picks off the first  $O$  eigenvalues,

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \begin{bmatrix} \mathbf{\Lambda}_O^{\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{\Lambda}_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 \right\}, \quad (4.59)$$

allowing us to further simplify the minimization by performing one more matrix multiplication,

$$\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left\| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right\|^2 \right\}. \quad (4.60)$$

To find the optimum method for selecting the eigenvectors to compose  $\mathbf{u}$ , we need to expand the vector magnitude operation in Eqn. (4.60) to give a minimum error,  $\min (\mathbf{e}^H \mathbf{R}_b \mathbf{e})$ , equal to

$$\min \left\{ \left( \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right)^H \left( \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right) \right\}. \quad (4.61)$$

Distributing the hermitian operation throughout the first quantity and using the fact that the eigenvalues are real<sup>1</sup> results in the error expression taking the form

$$\min \left\{ \left( \mathbf{r}_{bd}^H \mathbf{U} \Lambda^{-\frac{1}{2}} - \mathbf{r}_{bd}^H \mathbf{u} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} & \mathbf{0} \end{bmatrix} \right) \left( \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right) \right\}. \quad (4.62)$$

Further insight into the algorithm's error requires multiplying the two quantities as shown by

$$\begin{aligned} \min (\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min & \left\{ \mathbf{r}_{bd}^H \mathbf{U} \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} + \mathbf{r}_{bd}^H \mathbf{u} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right. \\ & \left. - \mathbf{r}_{bd}^H \mathbf{u} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} & \mathbf{0} \end{bmatrix} \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{r}_{bd}^H \mathbf{U} \Lambda^{-\frac{1}{2}} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right\}. \end{aligned} \quad (4.63)$$

---

<sup>1</sup>The eigenvalues are real because the eigenvalue decomposition was performed on the hermitian matrix  $\mathbf{R}_b$ .

Reducing this equation is a matter of recognizing we can substitute the vector magnitude,  $||^2$ , of each quantity. First, rearrange Eqn. (4.63) so that  $\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e})$  is equal to

$$\min \left\{ \left( \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right)^H \left( \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right) + \left( \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right)^H \left( \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right) - \mathbf{r}_{bd}^H \mathbf{u} \begin{bmatrix} \Lambda_O^{-1} & \mathbf{0} \end{bmatrix} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{r}_{bd}^H \mathbf{U} \begin{bmatrix} \Lambda_O^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{u}^H \mathbf{r}_{bd} \right\}. \quad (4.64)$$

The first two terms are recognized as the magnitude of a vector. The second two terms require some manipulation to see they are also vector magnitudes. The following shows the first step in the manipulation of these terms,

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 + \left| \Lambda_O^{-\frac{1}{2}} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 - \mathbf{r}_{bd}^H \mathbf{u} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \mathbf{U}^H \mathbf{r}_{bd} - \mathbf{r}_{bd}^H \mathbf{U} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \\ \mathbf{0} \end{bmatrix} \Lambda_O^{-\frac{1}{2}} \mathbf{u}^H \mathbf{r}_{bd} \right\}. \quad (4.65)$$

Distributing the eigenvector matrices  $\mathbf{U}$  and  $\mathbf{u}$  results in

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 + \left| \Lambda_O^{-\frac{1}{2}} \mathbf{u}^H \mathbf{r}_{bd} \right|^2 - \mathbf{r}_{bd}^H \begin{bmatrix} \mathbf{u} \Lambda_O^{-\frac{1}{2}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \mathbf{u}^H \\ \mathbf{0} \end{bmatrix} \mathbf{r}_{bd} - \mathbf{r}_{bd}^H \mathbf{u} \Lambda_O^{-\frac{1}{2}} \Lambda_O^{-\frac{1}{2}} \mathbf{u}^H \mathbf{r}_{bd} \right\}, \quad (4.66)$$

where we have taken advantage of the fact that  $\mathbf{U}$  was previously defined to have the form of  $\mathbf{U} = [\mathbf{U} \ \mathbf{C}]$ . This form of the equation shows the last two terms are also vector magnitudes,

$$\begin{aligned} \min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min & \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 + \left| \Lambda_O^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 \right. \\ & \left. - \left| \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \mathbf{U}^H \\ \mathbf{0} \end{bmatrix} \mathbf{r}_{bd} \right|^2 - \left| \Lambda_O^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 \right\}. \end{aligned} \quad (4.67)$$

Since the second and fourth terms cancel, the next operation necessary is to simply distribute the cross correlation vector  $\mathbf{r}_{bd}$  on the third term as shown by

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 - \left| \begin{bmatrix} \Lambda_O^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \\ \mathbf{0} \end{bmatrix} \right|^2 \right\}. \quad (4.68)$$

The second term simplifies even further because it is the magnitude of a vector. The  $\mathbf{0}$  matrix does not affect the magnitude of the vector and can be dropped resulting in

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \Lambda^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 - \left| \Lambda_O^{-\frac{1}{2}} \mathbf{U}^H \mathbf{r}_{bd} \right|^2 \right\}. \quad (4.69)$$

The final step merely substitutes the summation form of the vector products shown as the operands of the  $|\cdot|^2$  operation,

$$\min(\mathbf{e}^H \mathbf{R}_b \mathbf{e}) = \min \left\{ \left| \sum_{i=1}^{NM-1} \frac{\nu_i^H \mathbf{r}_{bd}}{\sqrt{\lambda_i}} \right|^2 - \left| \sum_{j=1}^O \frac{\nu_j^H \mathbf{r}_{bd}}{\sqrt{\lambda_j}} \right|^2 \right\}. \quad (4.70)$$

From this equation, the  $O$  eigenvectors comprising  $\mathbf{U}$  should be selected according to the metric

$$\left| \frac{\nu_i^H \mathbf{r}_{bd}}{\sqrt{\lambda_i}} \right|^2. \quad (4.71)$$



Choosing the  $O$  eigenvectors corresponding to the  $O$  maximum values of this metric will produce the minimum error in the reduced dimension algorithm, thereby producing the highest SINR.

This is the same result as shown by Goldstein's work [6], which he designates the *cross spectral metric*. Due to the use of this metric, the algorithm is referred to as the Cross Spectral Metric (CSM) algorithm for reduced dimension STAP. Obviously, this greatly differs from the principal components method. If the eigenvectors of  $\mathbf{R}_b$  are rank ordered vector according to Eqn. (4.71), then the selection of  $\mathbf{u}$  is simply a matter of selecting the first  $O$  eigenvectors.

The correlation between the desired signal and the noise projection vector  $\mathbf{r}_{bd}$  is interpreted as the cross-spectral energy. Therefore, the metric basically measures the cross-spectral energy projected along the  $i^{th}$  eigenvector. This energy relationship explains the squaring of the metric as shown in Eqn. (4.71). This energy comparison is an important consideration when trying to understand the advantages of the algorithm. The selection process solves the problem of inadvertently choosing an eigenvector along which the signal has no energy, i.e. the signal is naturally experiencing a null in this direction. Because there is a natural null, the noise has no effect on the signal. Expending valuable DOF creating a null in the antenna pattern to filter this noise is a waste. Creating antenna pattern nulls in directions of natural nulls is the problem inherent in the principal components method of eigenvector selection.

As validation of the mathematics, Fig. 4.6 compares the SINR for both methods using the interference plus noise covariance matrix  $\mathbf{R}$  from the previous examples. The expression for the SINR is developed in Chapter VI, see Eqn. (6.20). The figure is included here to illustrate the improvement in algorithm performance when the CSM is used.

Figure 4.6 shows the CSM method of choosing  $\mathbf{u}$  clearly outperforms the method of principle components. The algorithm rapidly converges to the optimum fully adaptive SINR and outperforms the principal components method at all dimensions up to the point where both reach the optimum SINR.

### SINR Comparison

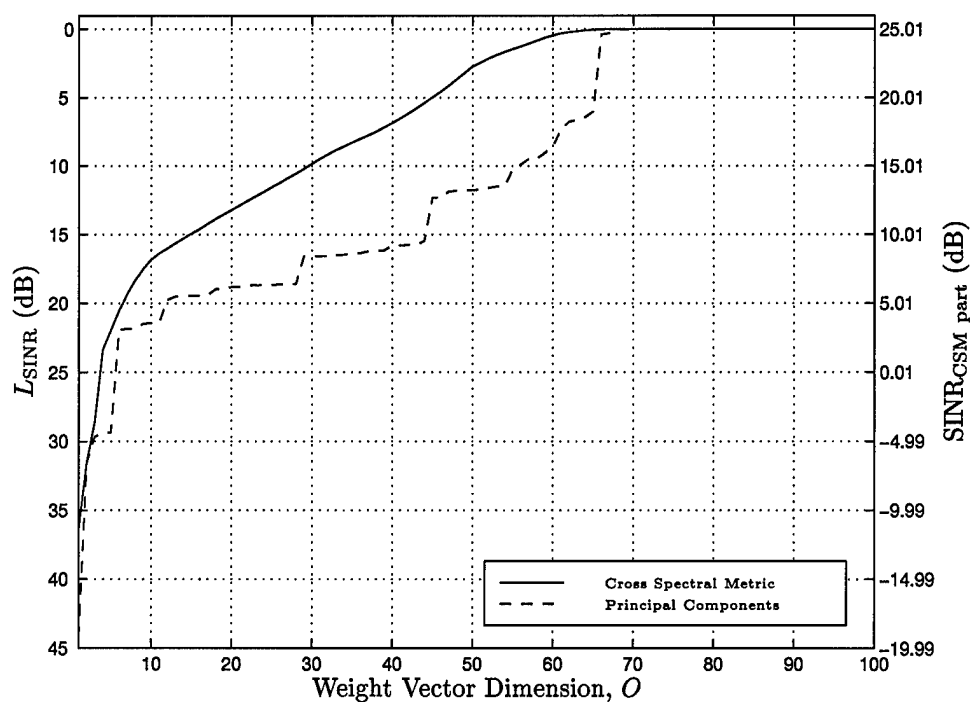


Figure 4.6 A comparison of the reduced dimensionality SINR for the principal components and cross spectral metric methods. The interference plus noise covariance matrix  $\mathbf{R}$  is the same as the one used to generate the SINR plots for Section 3.5. The cross spectral method results in a more rapid convergence to the optimum (fully adaptive) SINR than the principal components method.

In summary, Goldstein has shown the principal components method does not produce the optimum SINR for a reduced dimension adaptive processor. Instead, the CSM model as derived by Goldstein results in the maximum SINR.

#### 4.4 Summary

In order to gain insight into the problem of choosing the best DOF for reduced dimensionality STAP, the CSM model by Goldstein was examined in detail. This led to the cross-spectral metric, a method that results in the maximum SINR for a reduced dimension algorithm.

The basis for the cross spectral metric is the use of a subset of the eigenvectors of the interference plus noise covariance matrix  $\mathbf{R}_b$  as the reduced dimension projection matrix. The cross spectral metric then produces the optimum SINR for *this projection matrix*. If a different projection matrix is used, it is reasonable the SINR could be further improved.

The CSM algorithm is computationally expensive. The  $NM - 1$  dimension covariance matrix  $\mathbf{R}_b$  must first be determined. Further compounding the burden is the eigenvalue/eigenvector decomposition of this matrix, required to build the reduced dimension projection matrix  $\mathbf{U}$ . Savings in required secondary data support are not obvious because they come about through the Maximum Likelihood Estimator (MLE) used to estimate the covariance matrix  $\mathbf{R}$ .

If the secondary data support required for the CSM algorithm follows the work done by Reed [15], near optimum SINR performance should be obtained by only using  $2O$  secondary data vectors for the estimate of  $\mathbf{R}$ . Without the reduced dimensionality,  $NM$  secondary data vectors are required to reach optimum performance. Chapter V delves deeper into the MLE for  $\mathbf{R}$  and also introduces the problems associated with non-homogeneities in the secondary data used to estimate  $\mathbf{R}$ . Furthermore, a method of excising potential non-homogeneities is discussed and analyzed in detail for the FTS method. Chapter VI experimentally examines the CSM algorithm in terms of required secondary data support and the impact of non-homogeneities in the data.

## *V. Covariance Matrix Estimation and Non-homogeneity Detection*

### *5.1 Introduction*

One of the primary difficulties encountered in the implementation of a practical adaptive processor is estimation of the noise covariance matrix. All of the algorithms discussed rely on the interference plus noise covariance matrix  $\mathbf{R}$  for the adaptive weight calculation. Because of the rapidly changing environment encountered in an airborne platform, the noise statistics must be estimated from the data available. Furthermore, this estimate must be repeated for each range cell of interest. This repetition is necessary because the underlying assumption when examining each range cell is that there is a potential target within the range cell. This potential target represents a non-homogeneity in the data and should not be part of the estimate, even though a target does not necessarily exist at the range gate of interest.

The data can be broken up into two main classes, the primary data consisting of the range gate/cell of interest and the secondary data consisting of everything else. The secondary data is used for estimating the noise covariance matrix. Anything other than noise represents a non-homogeneity in the data. These non-homogeneities can range in severity from a surface clutter transition (e.g. land clutter to sea clutter) to a strong target in a range cell other than the one of interest.

In the adaptive filter developed to this point using the CSM algorithm or the development shown in Chapter III, the primary goal is to filter out the sources of interference. Any sources contained in the estimated covariance matrix will result in the adaptive filter expending Degrees Of Freedom (DOF) to build an antenna pattern with nulls on these sources. Some examples of interference sources desired in the covariance matrix include strong sources of clutter and barrage noise jamming. Conversely, the estimate should not include other targets. These targets waste valuable DOF building nulls with no real advantage in the detection of targets in the direction of interest.

## 5.2 Covariance Matrix Estimation

The interference plus noise covariance matrix  $\mathbf{R}$  is the foundation of the Wiener filter solution for the optimum weight vector  $\mathbf{w}$ . In Chapter III, this matrix was assumed known. In the examples shown in that chapter, it was estimated using the models developed in Sections 3.3.1, 3.3.2, and 3.3.3. However, the radar is now placed in the real world environment where all the parameters are constantly changing. The interference plus noise covariance matrix must be estimated from the data available.

The Maximum Likelihood Estimate (MLE)  $\hat{\mathbf{R}}$  for the interference plus noise covariance matrix is

$$\hat{\mathbf{R}}_l = \frac{1}{J} \sum_{i=l-J/2}^{l+J/2} \chi_i \chi_i^H \text{ for } i \neq l, \quad (5.1)$$

where the  $l$  subscript explicitly shows the estimate changes for each range gate,  $J$  is the number of secondary data vectors to be used in the estimate, and  $\chi_i$  is the  $i^{th}$  range gate space-time snapshot or data vector. As done previously in Chapter III, the dependence on range will be implicit from this point forward and the  $l$  subscript dropped. Obviously, the best estimate is obtained using the maximum amount of secondary data vectors. This maximum is equal to the number of range gates,  $L$ .

The interference covariance matrix  $\mathbf{R}$  in the Wiener solution or weight vector for all of the STAP algorithms is replaced by the estimate,  $\hat{\mathbf{R}}$ . This replacement for fully adaptive STAP is

$$\mathbf{w} = \hat{\mathbf{R}}^{-1} \mathbf{g}_t, \quad (5.2)$$

where  $\mathbf{g}_t$  is the tapered space-time steering vector to the target. If no taper is desired,  $\mathbf{v}_t$  is substituted.

An expression for the Signal-to-Interference-plus-Noise Ratio (SINR) is obtained by substituting Eqn. (5.2) into Eqn. (3.91) to obtain

$$\text{SINR} = \frac{\sigma^2 \xi_t \left| \mathbf{g}_t^H \hat{\mathbf{R}}^{-1} \mathbf{v}_t \right|^2}{\mathbf{g}_t^H \hat{\mathbf{R}}^{-1} \mathbf{R} \hat{\mathbf{R}}^{-1} \mathbf{g}_t}. \quad (5.3)$$

The problem with the use of this expression is the true interference covariance matrix  $\mathbf{R}$  is generally not known. This is the reason for using the MLE in the first place. However, an estimate for the true interference covariance matrix  $\mathbf{R}$  can be obtained by using all of the data vectors available to the MLE, whereas  $\hat{\mathbf{R}}$  would be obtained with some subset of these secondary data vectors.

The obvious question arising from the use of this MLE concerns the performance degradation. This performance degradation is directly related to the number of secondary data vectors,  $J$ , used in the MLE. Reed found this loss in SNR [15] as

$$L_{\text{SNR}} = -10 \log_{10} \left[ \frac{J + 2 - N_{\text{DOF}}}{J + 1} \right], \quad (5.4)$$

where  $N_{\text{DOF}}$  is the number of Degrees Of Freedom (DOF). Using this equation, performance within 3 dB of the optimal SNR given a known covariance matrix can be obtained by choosing  $J$  equal to [15]

$$J = 2N_{\text{DOF}} - 3 \approx 2N_{\text{DOF}}. \quad (5.5)$$

This is equivalent to a total secondary data set size of approximately  $2N_{\text{DOF}}$ . However, this result was developed by Reed for Sample Matrix Inversion (SMI) algorithms, i.e. the fully adaptive algorithm discussed in Chapter III. The nature of the FTS reduced dimension algorithm also allows the use of Eqn. 5.5. The particular point allowing the use of this sample size approximation is the FTS method computes the covariance matrix *after* the Doppler filtering operation. This filtering operation simply results in a new data set for the adaptive algorithm. From that point

onward, the computation of the weight vector is similar to that of a fully adaptive algorithm. The interference covariance matrix is estimated from the Doppler filtered data and then inverted in the process of forming the weight vector. The technique still falls under the SMI umbrella.

The CSM algorithm developed by Goldstein differs from the SMI techniques introduced by Ward [21] and Reed [2, 15]. As discussed in Section 4.3, the reduced dimension CSM algorithm requires the *full* dimension covariance matrix. The eigenvalue decomposition of this matrix is performed and then used to generate a reduced dimension covariance matrix. This matrix is then inverted in the computation of the optimum weight vector. The concerns with applying Eqn. 5.5 to the CSM reduced dimension algorithm are analyzed in detail in Chapter VI. This chapter focuses on understanding the MLE and its limitations.

A more fundamental problem with the MLE estimator is its assumption of independent, identically distributed (iid) data. This assumption is not generally valid. This is the reason for the non-homogeneity detector proposed by Melvin [14].

### 5.3 Non-Homogeneity Detection

As just explained, the MLE is based on the assumption the secondary data is iid. Generally, this assumption is not valid for real world data. Therefore, Melvin [14] introduced a method to select the vectors out of the entire set of secondary data that conform best to the iid assumption.

Melvin begins by defining homogeneous data. Given the covariance estimate from two different range cells  $i$  and  $j$ ,

$$\mathbf{R}_i = \mathcal{E}\{\chi_i \chi_i^H\} \text{ and } \mathbf{R}_j = \mathcal{E}\{\chi_j \chi_j^H\}, \quad (5.6)$$

then a homogeneous set of data satisfies [14]

$$\mathbf{R}_i \mathbf{R}_j^{-1} \approx \mathbf{I}_{NM}, \quad (5.7)$$

where  $NM \times NM$  is the dimension of the interference covariance matrix. The number of pulses in a Coherent Processing Interval (CPI) and number of antenna channels are denoted  $M$  and  $N$ , respectively.

Considering the other side of the problem, the secondary data vector is defined in terms of the components it contains,

$$\chi_i = \chi_{ci} + \chi_{ji} + \chi_{ni} + \chi_{oi}. \quad (5.8)$$

The components representing the homogeneous data are the homogeneous clutter component  $\chi_{ci}$ , white noise jamming  $\chi_{ji}$ , and uncorrelated noise  $\chi_{ni}$ . The non-homogeneous component is represented by  $\chi_{oi}$  and consists of colored noise jamming and the non-homogeneous clutter component.

Melvin proposes the Generalized Inner Product (GIP)

$$g_i = \chi_i^H \hat{\mathbf{R}}_s^{-1} \chi_i \quad (5.9)$$

as a method to determine a secondary data vector's degree of homogeneity. The matrix  $\hat{\mathbf{R}}_s$  represents an interference plus noise covariance matrix estimate based on the entire secondary data set. The GIP is then calculated for every vector contained within the secondary data set, producing the scalar  $g_i$ . The homogeneous data to be used in calculating the new interference covariance matrix estimate is then determined by choosing those vectors corresponding to

$$\bar{g} - \Delta_g \leq g_i \leq \bar{g} + \Delta_g, \quad (5.10)$$



where  $\bar{g}$  is the average of all the GIPs for the secondary data set and  $\Delta_g$  is chosen large enough such that only  $2O$  secondary data vectors satisfy the relationship. A good selection scheme might be to rank in ascending order the secondary data vectors according to the absolute value of their associated GIP with the mean GIP removed. The selection of the most homogeneous is then just a matter of choosing the first  $2O$  vectors.

To understand why the GIP works as a non-homogeneity detector, first rewrite Eqn. (5.9) in the form

$$\chi_i^H \hat{\mathbf{R}}_s^{-1} \chi_i = (\hat{\mathbf{R}}_s^{-\frac{1}{2}} \chi_i)^H \hat{\mathbf{R}}_s^{-\frac{1}{2}} \chi_i = \left| \hat{\mathbf{R}}_s^{-\frac{1}{2}} \chi_i \right|^2, \quad (5.11)$$

where the square root of the matrix can be determined through its eigenvalue/eigenvector decomposition. The final term shown is the output power of a whitening filter operating on the data vector. If we denote the output of the whitening filter as  $\tilde{\chi}_i$ , then the covariance matrix of this output is

$$\tilde{\mathbf{R}}_i = \mathcal{E} \{ \tilde{\chi}_i \tilde{\chi}_i^H \} = \mathcal{E} \left\{ \hat{\mathbf{R}}_s^{-\frac{1}{2}} \chi_i \chi_i^H \hat{\mathbf{R}}_s^{-\frac{1}{2}} \right\} = \hat{\mathbf{R}}_s^{-\frac{1}{2}} \mathbf{R}_i \hat{\mathbf{R}}_s^{-\frac{1}{2}}, \quad (5.12)$$

where  $\mathbf{R}_i$  represents the true covariance matrix of the data vector under test,  $\chi_i$ . Examining this equation shows that as the covariance matrix estimate  $\hat{\mathbf{R}}_s$  approaches the true covariance matrix  $\mathbf{R}_i$ , then

$$\tilde{\mathbf{R}}_i \rightarrow \mathbf{I}_{NM} \text{ as } \hat{\mathbf{R}}_s \rightarrow \mathbf{R}_i. \quad (5.13)$$

This relationship further solidifies the whitening filter concept. If the data is actually correlated according to the estimate  $\hat{\mathbf{R}}_s$ , then the data is successfully whitened to a zero mean unit variance

random vector. Hence, the expected value of the GIP becomes

$$\mathcal{E}\{g_i\} = \mathcal{E}\left\{\chi_i^H \hat{\mathbf{R}}_s^{-1} \chi_i\right\} = \mathcal{E}\left\{\tilde{\chi}_i^H \tilde{\chi}_i\right\}, \quad (5.14)$$

where  $\tilde{\chi}_i$  is the whitened data. Since this data is now white and the data samples are assumed independent, then the inner product shown is the sum of the unit variances. Hence,

$$\mathcal{E}\{g_i\} = \mathcal{E}\left\{|\tilde{\chi}_i[0]|^2 + |\tilde{\chi}_i[1]|^2 + \dots + |\tilde{\chi}_i[NM - 1]|^2\right\} = NM. \quad (5.15)$$

Any deviation from this value flags a non-homogeneity in the data.

Using this new knowledge about the homogeneity of the data, the MLE estimator for the  $l^{th}$  range gate can be rewritten as

$$\hat{\mathbf{R}}_l = \frac{1}{J} \sum_{\substack{i \in \Psi \\ i \neq l}} \chi_i \chi_i^H, \quad (5.16)$$

where  $\Psi$  is the set of  $J$  vectors satisfying Eqn. (5.10). Although the above formula only stipulates  $i \neq l$ , typically two guard cells on each side of the range cell of interest are also removed from the secondary data used in the interference covariance matrix calculation.

#### 5.4 MCARM Example

To illustrate the effectiveness of the non-homogeneity detector, an example is offered from the MCARM database. See the MCARM reports [18] for a description of the database and the flight tests used to generate the data.

The data file used for this example is rd050465. Because of the limited size of the Coherent Processing Interval (CPI), the Factored Time-Space (FTS) algorithm was used. The number of

range cells in the CPI supports the sample size requirement for the interference covariance matrix estimate  $\hat{\mathbf{R}}$ .

A Blackman Harris window [11] was used for the Doppler filter. This window possesses excellent sidelobe level characteristics. The first sidelobe level is -92 dB with -6 dB/Octave roll-off from the first sidelobe [11].

The output of the Doppler filter is two dimensional. There is a Doppler filter bank for every range cell. Within each bank there are  $M$  bins, equal to the number of pulses in the CPI. Although, the output of the filter could be examined with a three dimensional plot, many of the important characteristics are not easily visible. For this reason, two figures are presented to describe the output of the Doppler filter.

The Doppler filtering operation is performed for *each* antenna element. Therefore, there are  $N$  Doppler filters each with a two dimensional output. If we wanted to form a beam perpendicular to the antenna array (boresight), we could simply add the output of the Doppler filters together. However, the transmit direction for this data file is not boresight, it is  $19.155^\circ$  (reference the MCARM data manuals for the coordinate system [18]). Hence, to examine the output of the Doppler filter at the target location we must form a beam in the direction of the target. The target location was conveniently chosen to correspond to the transmit direction of the radar. The beamforming operation is exactly the same as that described in Chapter II. There is no spatial windowing used in the non-adaptive beamforming of the Doppler filters.

Figure 5.1 shows the output of the Doppler filter bank corresponding to range cell 450. The dimension is across the  $M$  bins of the bank (there are 128 pulses in the CPI and hence 128 bins in the Doppler filter bank). The filtering operation was performed on both the original data file and the data file with an artificial target inserted. The 22 (there are 22 antenna channels in the CPI) Doppler filters were beamformed to the transmit and target azimuth of  $19.155^\circ$  with no spatial windowing used. Table 5.1 presents the parameters of the target. The parameter of interest for

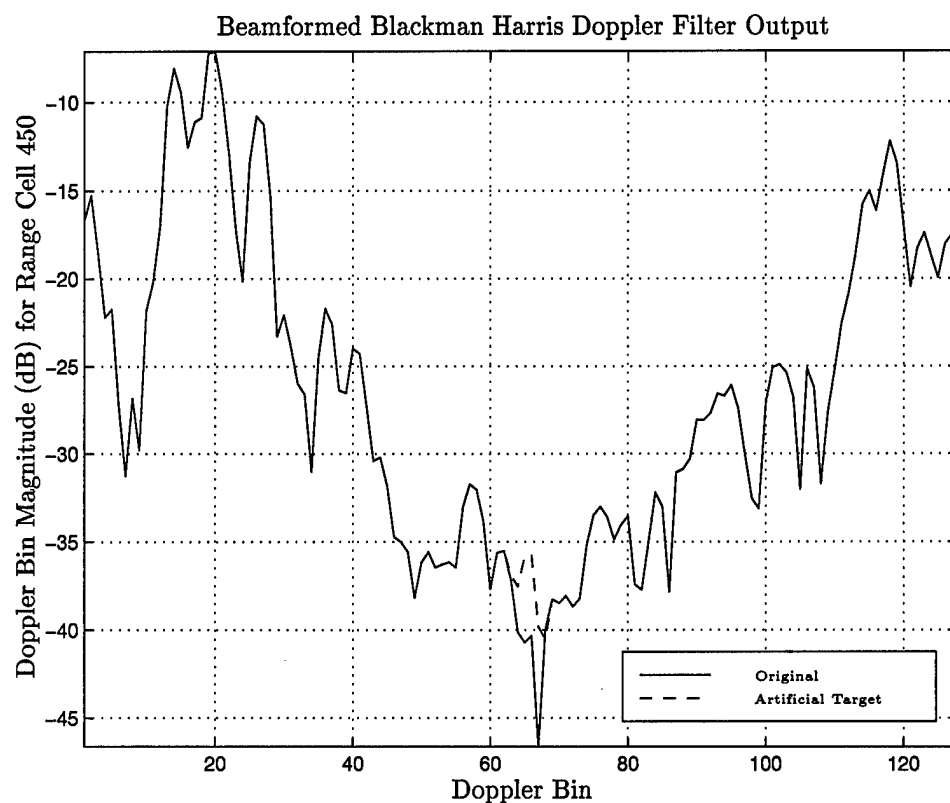


Figure 5.1 The Blackman Harris Doppler filter output beamformed to the transmit angle of  $19.155^\circ$  for range cell 450. The artificial target is located at Doppler bin 65. The effect of inserting the target into the data is clearly visible by comparing the two curves.

Table 5.1 Target parameters used for the GIP non-homogeneity detector example.

Variable	Value
Range Cell	450
Doppler Frequency	1000 Hz
Doppler Bin	65
Azimuth	19.155°

this plot is the Doppler bin location of the target, bin 65. Examining the figure shows the effect of inserting the target into the data, a small spike at bin 65. Without the original data curve as a reference, it is impossible to determine the target exists at this point.

The second dimension of the Doppler filter output is across the  $L$  (the number of range cells, 630) filter banks. We are only interested in examining the output of bank 450, corresponding to the location of the target at range cell 450. Rather than plot across all 630 range cells, a small subset is chosen. In this case the subset is from cells 420 to 490. Because the target is located at Doppler bin 65, this is the bin chosen for the plot in Fig. 5.2. Both the original data and the data with the artificial target is plotted. Without the original data as a reference, detection of the target is not possible at this point.

The amplitude of the target was chosen to highlight the advantage of an adaptive filtering/beamforming operation. Up to this point in the example, the presence of the target cannot be detected. Typically, a detection scheme would use Fig. 5.2 as a method to determine target presence. A threshold is set and target presence declared if the output amplitude of the Doppler bin of interest at the range cell of interest exceeds this threshold. At this point in the example, we have chosen an exceptional Doppler filter window (the Blackman Harris) and beamformed in the direction of the target. Yet we still cannot verify its presence. The only option left is windowing in the spatial domain (in the non-adaptive beamforming operation). However, for comparison purposes to the adaptive FTS algorithm, spatial windowing will not be used in the example.

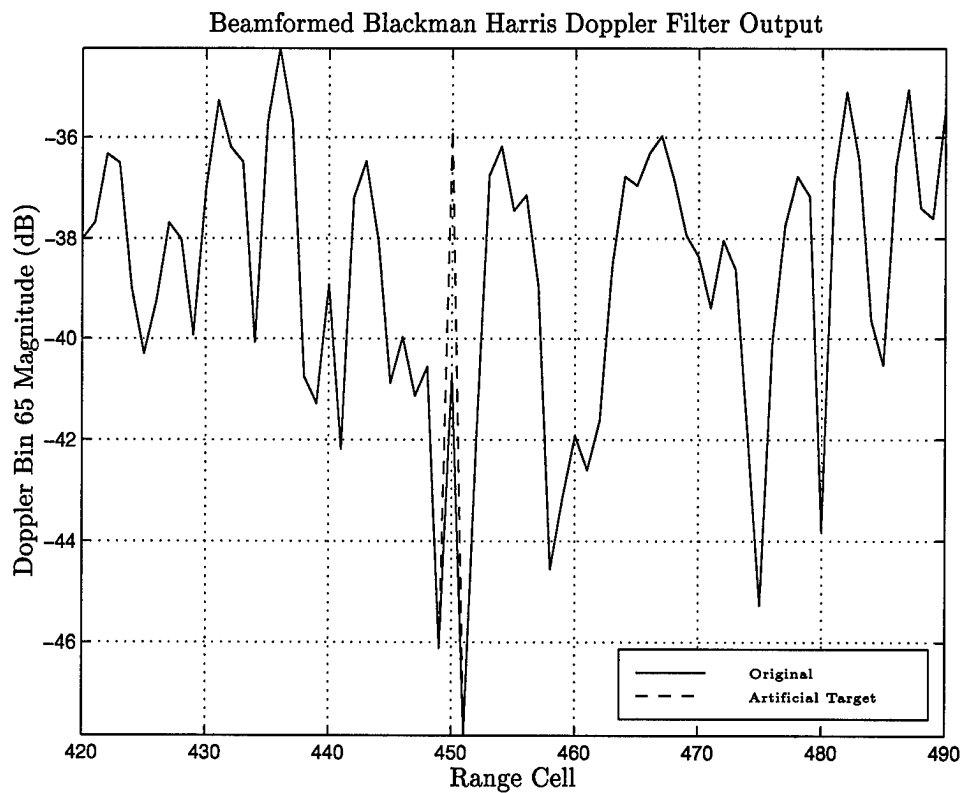


Figure 5.2 The Blackman Harris Doppler filter output beamformed to the transmit azimuth of  $19.155^\circ$  for Doppler bin 65. The artificial target inserted at range cell 450 is not threshold detectable at this point, there are other peaks of higher amplitude. However, given the original data as a reference (the solid line), the target is clearly visible.

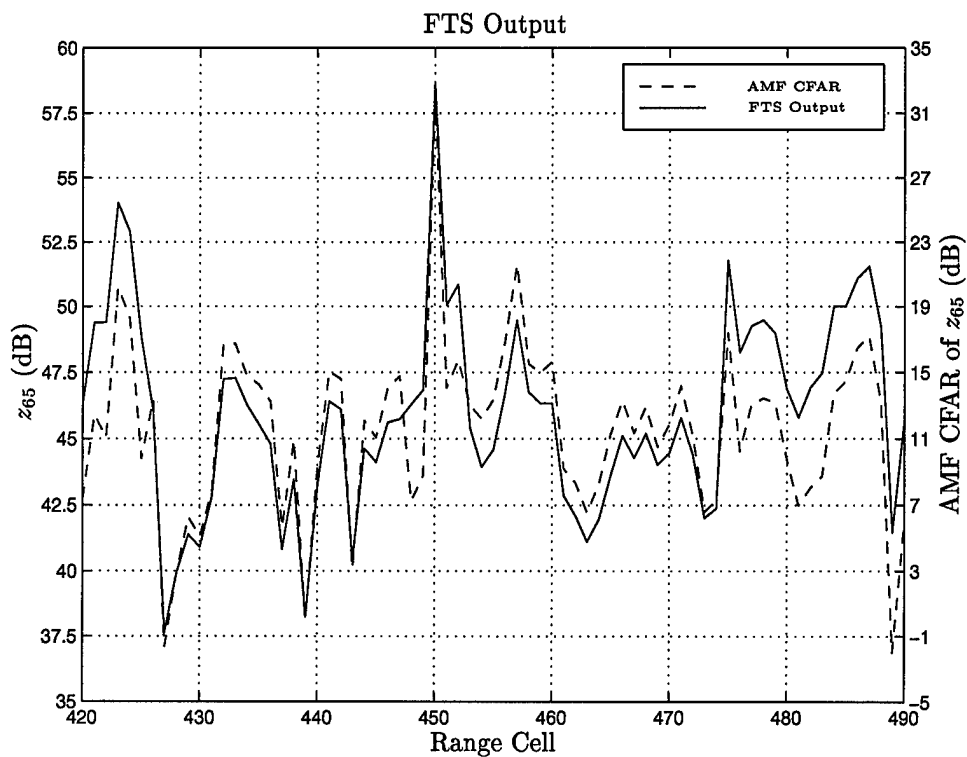


Figure 5.3 The FTS (with Blackman Harris Doppler filter) algorithm output. The target inserted at range cell 450, Doppler bin 65 is distinctly shown. The AMF CFAR test statistic, shown by the dashed line, illustrates a distinct improvement over the raw FTS output. This statistic also allows the use of a constant threshold based on probability of false alarm.

Now let's implement the FTS algorithm and examine the output for the target. See Section 3.5 for a discussion of FTS. No spatial windowing is used in the implementation of the algorithm to allow easy comparison to the non-adaptive plots just presented. The Doppler filter is the same as that used above, the Blackman Harris. The number of secondary data vectors required (to give an output within 3 dB of optimum [15]) for the covariance matrix estimate is twice the number of DOF. Since the FTS algorithm is adaptive in the spatial domain only, there are 22 DOF (11 over 11 channel array) and 44 secondary data vectors are required. For the FTS output shown in Fig. 5.3, these 44 vectors are chosen symmetrically about the range cell under test (using the same Doppler bin from each range cell), neglecting two guard cells on each side. For example, the covariance matrix estimate for range cell 450 and Doppler bin 65 (the target cell and bin) will use the secondary data vectors from Doppler bin 65 and range cells 426-447, 453-474.

As Fig. 5.3 shows, the target is clearly visible at range cell 450. In fact, it is approximately 5 dB larger than the next highest peak. Threshold detection would work well here, resulting in target detection. The problem with the raw FTS output is it does not possess a Constant False Alarm Rate (CFAR) property. For this reason, the output of the algorithm depends on the noise and a constant threshold cannot be set. Therefore, several test statistics have been introduced possessing the desirable CFAR property. This property allows a single threshold once probability of false alarm has been chosen. See Appendix C for a discussion of these test statistics and more details on the theory behind them. For this example, the Adaptive Matched Filter CFAR (AMF-CFAR) test was chosen. It is also plotted in Fig. 5.3, its amplitude corresponds to the scale on the right side of the graph. The embedded CFAR characteristic also results in a greater separation between the target peak and the next lower peak. It is increased from approximately 5 dB to greater than 10 dB.

Now let us examine the output of the FTS algorithm when the GIP non-homogeneity detector is used. First, the interference covariance matrix is determined from the full set of secondary data.



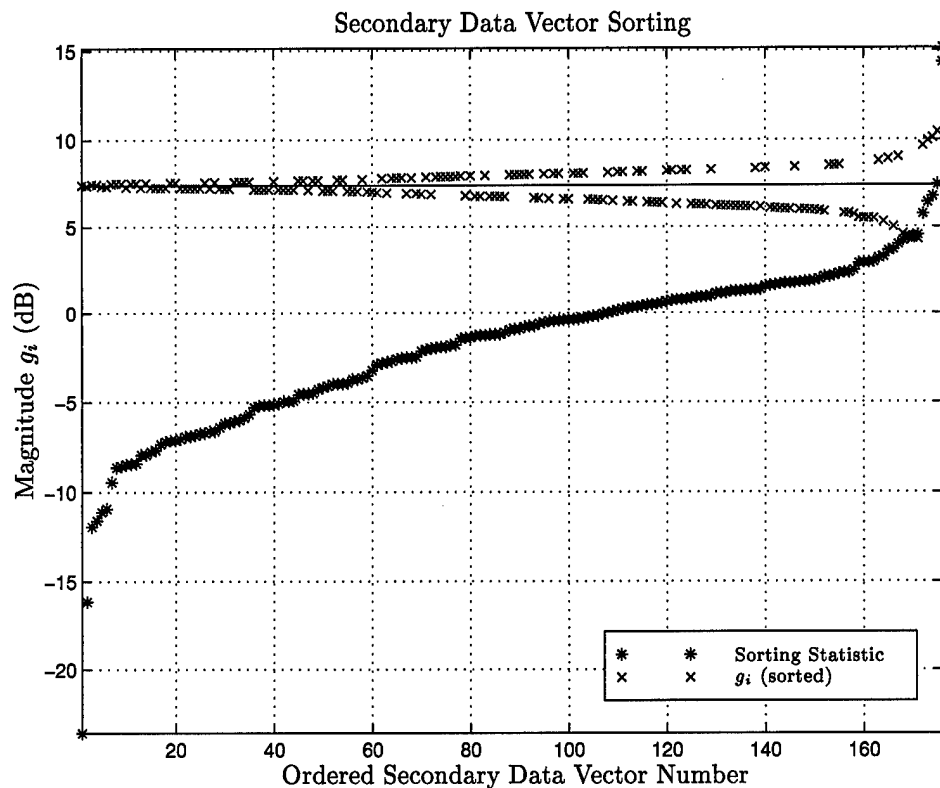


Figure 5.4 The GIP and sorting statistic used to order the secondary data vectors according to homogeneity. The most homogeneous vectors are closest to the mean (the horizontal line). The sorting statistic is shown in Eqn. (5.17). This plot illustrates the sorting statistic results in the correct ordering of the secondary data vectors by GIP magnitude.

In this case, the secondary data set size was set to four times the required number of secondary data vectors. Therefore, the secondary data set size was set to 176. These 176 data vectors are chosen symmetrically about the range cell under test. Figure 5.4 shows the ordering performed using the GIP. The test statistic used to order the vectors was

$$\text{sorting statistic} = |g_i - \bar{g}|, \quad (5.17)$$

where  $g_i$  is the GIP for the  $i^{\text{th}}$  secondary data vector and  $\bar{g}$  is the mean of all the GIPs within the secondary data set. The figure shows both the test statistic and the true magnitude of the GIP. The test statistic does result in the correct ordering of the data vectors, since the most homogeneous are clustered about the mean. Choosing the 44 secondary data vectors according to this ordering results in those closest to the mean being chosen first as desired. Selection is merely a matter of choosing the first 44 secondary data vectors.

Figure 5.5 is an alternate way of viewing the sorting process. The  $x$ -axis of the plot shows the sorted data vector number. This number is the order in which the data vectors are used in the MLE for the interference covariance matrix estimate. The  $y$ -axis is the corresponding original data vector number in relation to the range cell under test. The value of 0 corresponds to the range cell under test, hence this data vector is not used in the MLE as illustrated by the absence of an asterisk along this line in the plot. The data in the plot is for range cell 490. Similarly, the two guard cells on each side of the range cell under test are also not used in the MLE. As the figure shows, the use of GIP ordering results in a drastically different set of secondary data vectors, since the first 44 of the ordered set are chosen to generate the new interference covariance matrix  $\mathbf{R}$  estimate. Also, the target in range cell 450 represents the most severe non-homogeneity in the data, hence it is chosen last as indicated by the arrow in Fig. 5.5.

Figure 5.6 shows the output of the FTS algorithm when using the GIP non-homogeneity detector. The target at range cell 450 is clearly visible and the difference between the target

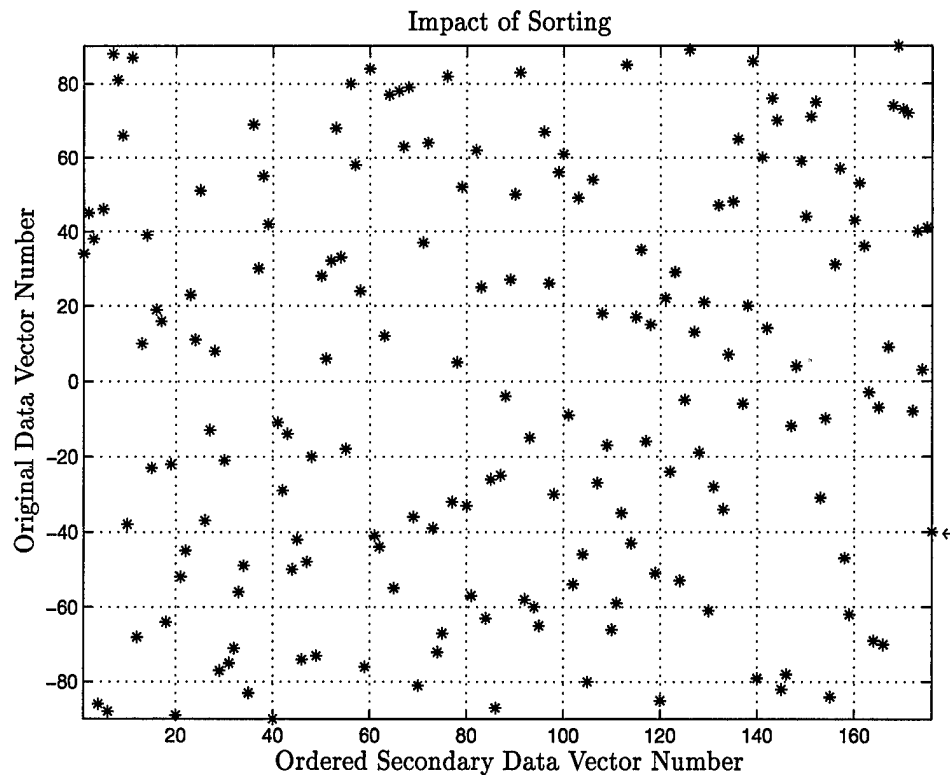


Figure 5.5 A different viewpoint of the ordering process. The sorting data is from the interference covariance matrix estimate for range cell 490. Cell 490 corresponds to 0 on the  $y$ -axis. This figure illustrates the impact of sorting the data vectors according to homogeneity. As expected, the target represents the most severe non-homogeneity and hence it is chosen last (data point shown by the arrow corresponds to cell  $490-40=450$ ).

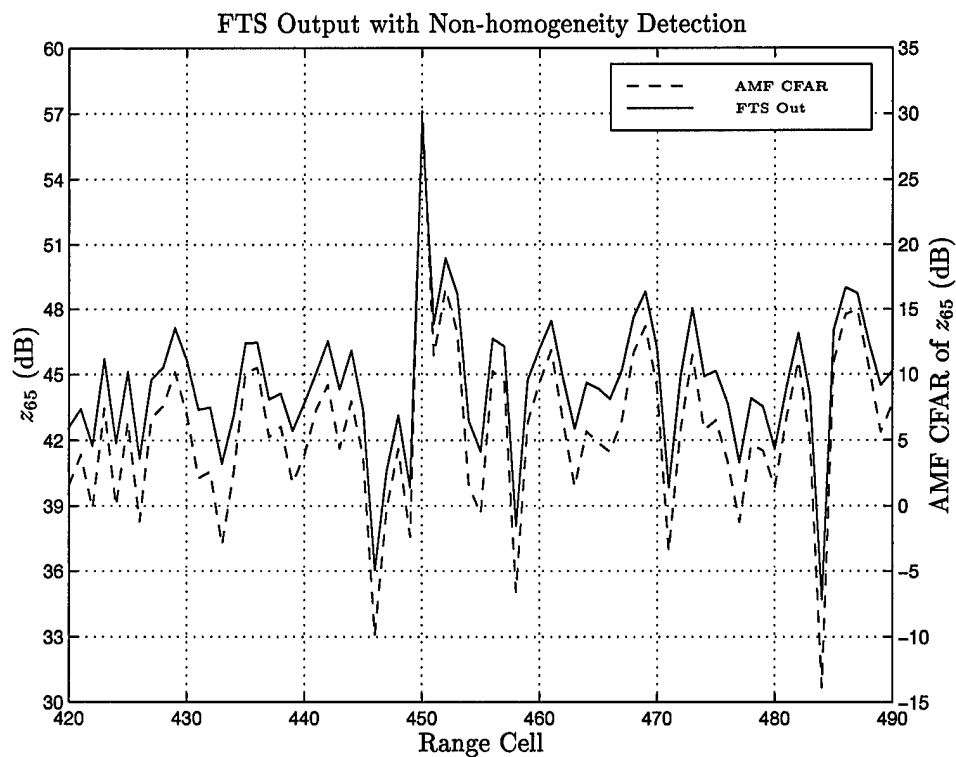


Figure 5.6 The output of the FTS algorithm for Doppler bin 65 when using the GIP non-homogeneity detector. The target is clearly shown at range cell 450 with a difference of 7 dB to the next highest peak. The AMF CFAR statistic increases this difference to 13 dB.

peak and noise floor is considerably larger than without non-homogeneity detection. Figure 5.6 also shows the output of the FTS algorithm when using the AMF CFAR statistic. Even more separation is obtained between the target peak and the noise floor.

### 5.5 Summary

This chapter introduced the underlying architecture behind determining the interference covariance matrix estimate  $\mathbf{R}$  from real world data. The estimate is obtained through the use of the MLE. However, the primary limitation to the MLE is the assumption of iid data. This assumption is not generally valid in the real world.

Melvin's GIP was introduced as a method to detect non-homogeneities in the secondary data. The philosophy behind its operation was discussed in detail. As proof of its operation, an example using the MCARM database is shown in Section 5.4. The figures shown in this section illustrate the improvement obtained by using this non-homogeneity detector.

Although the example presented in this chapter shows considerable improvement, an even greater improvement is obtained if a large second target is placed in the data. This second target corrupts the interference covariance matrix estimate for the range cell of the first, making the original target undetectable even using the AMF CFAR test statistic. Using the GIP non-homogeneity detector results in a target detection.

The most significant drawback to the use of the GIP as a non-homogeneity detector is the computational cost associated. The interference covariance matrix must first be estimated for the entire secondary data set. Upon completion of this estimate, all of the secondary data vectors must be rank ordered. The final version of  $\hat{\mathbf{R}}$  is obtained using only the most homogeneous members of the secondary data set. The computational cost associated with this is immense. It severely offsets the gains associated with implementing the reduced dimension algorithm.

## *VI. Analysis of the CSM Algorithm*

### *6.1 Introduction*

Previous chapters have explored the concepts of Space-Time Adaptive Processing (STAP) and extended the material through many examples and MATLAB® simulations. The thesis now focuses on satisfying the initial goals, determination of the sample support required for the Cross Spectral Metric (CSM) algorithm and its susceptibility to non-homogeneities in the secondary data.

A second issue addressed is the question of sample support for the Maximum Likelihood Estimate (MLE) of the interference plus noise covariance matrix,  $\hat{\mathbf{R}}$ . As discussed briefly in Chapter V, the analysis done by Reed [15] applies to STAP methods using Sample Matrix Inversion (SMI). There are distinct differences between the SMI techniques and the CSM algorithm that bring into question the application of Reed's analysis.

This chapter is organized into three main sections. The first section analyzes the performance of the CSM algorithm in terms of the Signal-to-Interference-plus Noise Ratio (SINR). The analysis is done for a known interference plus noise covariance matrix,  $\mathbf{R}$ . This analysis gives optimum performance curves, where optimum signifies the curves represent an SINR upper bound for the  $\mathbf{R}$  of interest.

These curves serve as a foundation for comparing the results of the next section. This second section attempts to define the secondary data support required in the MLE to obtain a certain threshold of the optimum SINR performance. The analysis is done through the use of Monte Carlo simulation.

The third section of this chapter examines the effects of non-homogeneities in the secondary data. As shown in Chapter V, the Factored Time Space (FTS) algorithm is susceptible to non-homogeneities in the secondary data. Here, the susceptibility of the CSM algorithm is examined and the Generalized Inner Product (GIP) non-homogeneity detector [14] is implemented with this algorithm.

## 6.2 SINR Comparisons for Known Covariance

**6.2.1 Fully Adaptive.** Let us begin the analysis of the CSM algorithm through an examination of SINR with known interference plus noise covariance  $\mathbf{R}$ . The direct form and FTS algorithms of Chapter III are used for comparison.

From Eqn. (3.91), the SINR in terms of the full dimensioned weight vector and  $\mathbf{R}$  is

$$\text{SINR} = \frac{\sigma^2 \xi_t |\mathbf{w}^H \mathbf{v}_t|^2}{\mathbf{w}^H \mathbf{R} \mathbf{w}}, \quad (6.1)$$

where  $\sigma^2$  is the noise power per element,  $\xi_t$  is the target SNR per element, and  $\mathbf{v}_t$  is the unnormalized space-time steering vector. This equation can be applied to the CSM algorithm if the weight vector describing the entire model is developed.

Deriving a weight vector that fully describes the CSM algorithm is rather simple. We will begin with the full dimension model. Begin by writing the output of the full dimensional CSM algorithm as

$$y = d - y_b, \quad (6.2)$$

where  $y$  is the output of the CSM algorithm,  $d$  is the output of the upper branch, and  $y_b$  is the output of the lower branch. Figure 4.3 depicts the model we are currently working with. From the development in Chapter IV and the block diagram in Fig. 4.3, we can write Eqn. (6.2) as

$$\begin{aligned} y &= \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_b^H \mathbf{b} \\ &= \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_b^H \mathbf{B} \boldsymbol{\chi} \\ &= (\mathbf{s}^H - \mathbf{w}_b^H \mathbf{B}) \boldsymbol{\chi}, \end{aligned} \quad (6.3)$$

where  $\mathbf{w}_b$  is the  $MN - 1$  dimensional weight vector and  $\mathbf{s}$  is the normalized space-time steering vector, both from Section 4.3. The equation now fits the form of the standard fully adaptive algorithm  $\mathbf{y} = \mathbf{w}^H \boldsymbol{\chi}$  with the hermitian of the full dimensioned weight vector as

$$\mathbf{w}^H = \mathbf{s}^H - \mathbf{w}_b^H \mathbf{B}. \quad (6.4)$$

Substituting Eqn. (4.16) for  $\mathbf{w}_b$  and taking the hermitian of both sides results in the following form for the full dimension weight vector,

$$\mathbf{w} = \mathbf{s} - \mathbf{B}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd}. \quad (6.5)$$

We can now substitute this weight vector description of the fully adaptive CSM algorithm into the SINR expression of Eqn. (6.1). This substitution results in  $\text{SINR}_{\text{CSM full}}$  given by

$$\text{SINR}_{\text{CSM full}} = \frac{\sigma^2 \xi_t \left| (\mathbf{s} - \mathbf{B}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd})^H \mathbf{v}_t \right|^2}{(\mathbf{s} - \mathbf{B}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd})^H \mathbf{R} (\mathbf{s} - \mathbf{B}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd})}, \quad (6.6)$$

where the subscripted  $\text{SINR}_{\text{CSM full}}$  is to distinguish this expression from future expressions.

Figure 6.1 shows a plot of Eqn. (6.6) versus normalized Doppler  $\bar{\omega}$ . For comparison to the SINR plots shown in Chapter III, both  $\sigma^2$  and  $\xi_t$  have been set to 1 and the interference plus noise covariance matrix  $\mathbf{R}$  was modeled using the same scenario as the examples in Chapter III. This scenario includes two noise jammers (located at  $-40^\circ$  and  $25^\circ$ ), clutter, and noise. The antenna array is composed of 18 elements and 18 pulses were used in the CPI.

It is important to point out some potential pitfalls in using Eqn. (6.6). When plotting the equation, the steering vectors  $\mathbf{s}$  (normalized) and  $\mathbf{v}$  (un-normalized) are not the only parameters changing with normalized Doppler  $\bar{\omega}$  (and/or spatial frequency  $\vartheta$  if a plot of angle is desired also). Because the blocking matrix  $\mathbf{B}$  is defined such that  $\mathbf{B}\mathbf{s} = \mathbf{0}$ , then every time  $\mathbf{s}$  changes the blocking



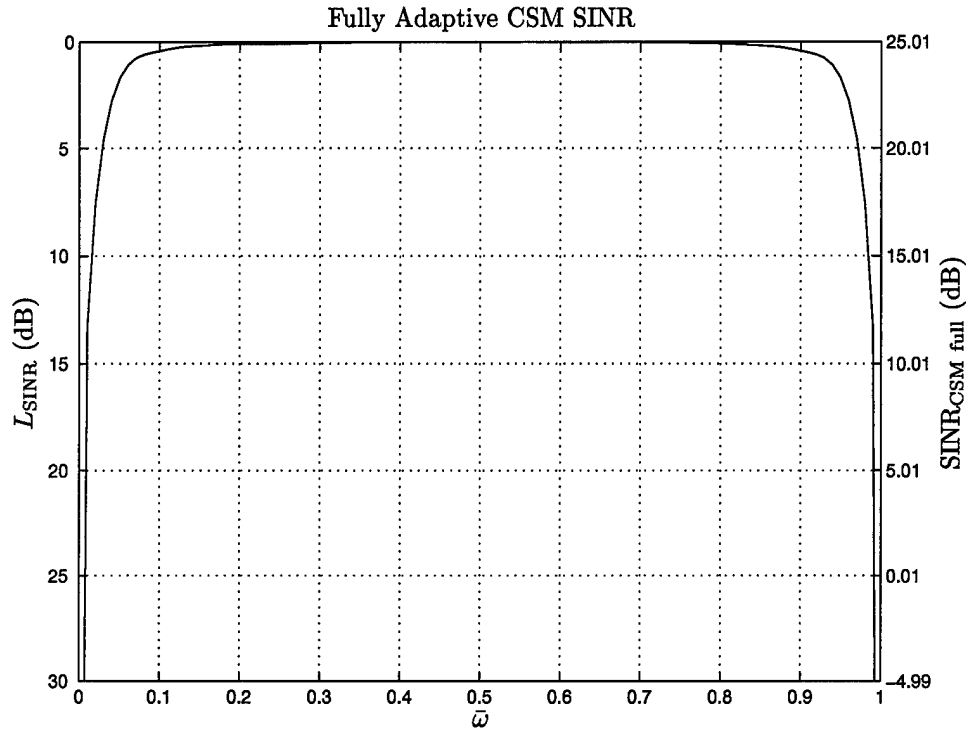


Figure 6.1 A plot of the fully adaptive SINR expression of the CSM algorithm versus normalized Doppler. The interference plus noise covariance matrix  $\mathbf{R}$  is the same as the one used to generate the SINR plots in Section 3.5 (the CPI includes 18 pulses and 18 antenna elements with  $\mathbf{R}$  including effects from two jammers, noise, and clutter). As expected, this plot of the CSM algorithm fully adaptive SINR matches the model from Chapter III.

matrix changes also. Therefore, it is also dependent on  $\bar{\omega}$ . Furthermore, the interference plus noise covariance matrix projection  $\mathbf{R}_b$  is equal to  $\mathbf{B}\mathbf{R}\mathbf{B}$  and cross correlation vector  $\mathbf{r}_{bd}$  is equal to  $\mathbf{B}\mathbf{R}\mathbf{s}$ . Hence,  $\mathbf{R}_b$  and  $\mathbf{r}_{bd}$  are also dependent on  $\bar{\omega}$ . This makes the calculation of  $\text{SINR}_{\text{CSM full}}$  very intensive as the inverse of  $\mathbf{R}_b$  must be recalculated at each  $\bar{\omega}$ .

All of these pitfalls have been avoided in Fig. 6.1. As the figure depicts, the CSM fully adaptive model produces results very similar to the model developed in Chapter III. The only difference is the peak SINR of 25.01 dB, where the previous model generated a peak of 25.1 dB. The difference is likely due to rounding errors incurred in the more complex SINR expression for this model.

The same SINR examination can be computed for the full dimension CSM algorithm with eigenvector projection matrix  $\mathbf{U}$  shown in Fig. 4.4. Examination of the figure shows the output of the model can be written in the same manner as the above work,

$$y = d - y_p. \quad (6.7)$$

The upper branch of the model is exactly the same as before, the projection of the incoming data vector on the normalized steering vector to give

$$y = \mathbf{s}^H \boldsymbol{\chi} - y_p. \quad (6.8)$$

The lower branch output is the inner product of  $\mathbf{p}$  and the weight vector,

$$y = \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_p^H \mathbf{p}. \quad (6.9)$$

Working backwards through the block diagram builds the expression allowing us to pull out the full dimension weight vector,

$$\begin{aligned} y &= \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_p^H \mathbf{U}^H \mathbf{b} \\ &= \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_p^H \mathbf{U}^H \mathbf{B} \boldsymbol{\chi}. \end{aligned} \quad (6.10)$$

The expression for the weight vector  $\mathbf{w}_p$  was previously introduced in Eqn. (4.21),

$$\mathbf{w}_p = \mathbf{U}^H \mathbf{w}_b. \quad (6.11)$$

Substituting this expression results in

$$y = \mathbf{s}^H \boldsymbol{\chi} - \mathbf{w}_b^H \mathbf{U} \mathbf{U}^H \mathbf{B} \boldsymbol{\chi}. \quad (6.12)$$

Since the eigenvector projection matrix  $\mathbf{U}$  is unitary, the product  $\mathbf{U} \mathbf{U}^H$  is the identity matrix.

Furthermore, substituting for the weight vector  $\mathbf{w}_b$  produces

$$y = \mathbf{s}^H \boldsymbol{\chi} - \mathbf{r}_{bd}^H \mathbf{R}_b^{-1} \mathbf{B} \boldsymbol{\chi}. \quad (6.13)$$

Factoring out the incoming space-time snapshot (or data vector)  $\boldsymbol{\chi}$  shows this expression is the same as that derived in Eqn. (6.5),

$$y = (\mathbf{s}^H - \mathbf{r}_{bd}^H \mathbf{R}_b^{-1} \mathbf{B}) \boldsymbol{\chi} \quad (6.14)$$

which implies that

$$\mathbf{w} = \mathbf{s} - \mathbf{B}^H \mathbf{R}_b^{-1} \mathbf{r}_{bd}. \quad (6.15)$$

The two forms of the full dimension CSM algorithm have been shown equivalent in SINR.

**6.2.2 Partially Adaptive.** We have gained insight into the CSM algorithm through a comparison of its fully adaptive SINR to the fully adaptive SINR from the model developed in Chapter III. Let's gain further insight by examining the SINR for the reduced dimension model.

Referencing Fig. 4.5 and Eqn. (4.30), the output of the reduced dimension CSM algorithm can be written as

$$y_r = d - y_z = d - \mathbf{w}_z^H \mathbf{U}^H \mathbf{b}. \quad (6.16)$$

Substituting for  $d$  and  $\mathbf{b}$  in the above equation and factoring out the space-time snapshot  $\chi$  results in the form needed to formulate a full dimension weight vector describing the entire CSM algorithm,

$$y_r = \left( \mathbf{s}^H - \mathbf{w}_z^H \mathbf{u}^H \mathbf{B} \right) \chi. \quad (6.17)$$

From this equation, the full dimension weight vector for the partially adaptive CSM algorithm is

$$\mathbf{w} = \mathbf{s} - \mathbf{B}^H \mathbf{u} \mathbf{w}_z. \quad (6.18)$$

Furthermore, we can substitute Eqn. (4.26) to obtain

$$\mathbf{w} = \mathbf{s} - \mathbf{B}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd}, \quad (6.19)$$

where  $\Lambda_O$  is the diagonal matrix of eigenvalues corresponding to the  $O$  eigenvectors used to compose  $\mathbf{u}$ . These eigenvectors are chosen through two popular methods, the cross spectral metric as developed in Chapter IV and the principal components method. The principal components method simply chooses the eigenvectors according to the largest eigenvalue magnitude. However, as shown in Chapter IV this method does not result in the optimum SINR.

The next step is to substitute the full dimension weight vector for the partially adaptive CSM algorithm into Eqn. (6.1). This substitution results in  $\text{SINR}_{\text{CSM part}}$  given by

$$\text{SINR}_{\text{CSM part}} = \frac{\sigma^2 \xi_t \left| \left( \mathbf{s} - \mathbf{B}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right)^H \mathbf{v}_t \right|^2}{\left( \mathbf{s} - \mathbf{B}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right)^H \mathbf{R} \left( \mathbf{s} - \mathbf{B}^H \mathbf{u} \Lambda_O^{-1} \mathbf{u}^H \mathbf{r}_{bd} \right)}. \quad (6.20)$$

The same pitfalls exist here as in the equation for the fully adaptive CSM algorithm. The parameters  $\mathbf{s}$ ,  $\mathbf{v}$ ,  $\mathbf{B}$ ,  $\mathbf{u}$ ,  $\Lambda_O$ , and  $\mathbf{r}_{bd}$  all depend on  $\bar{\omega}$ .

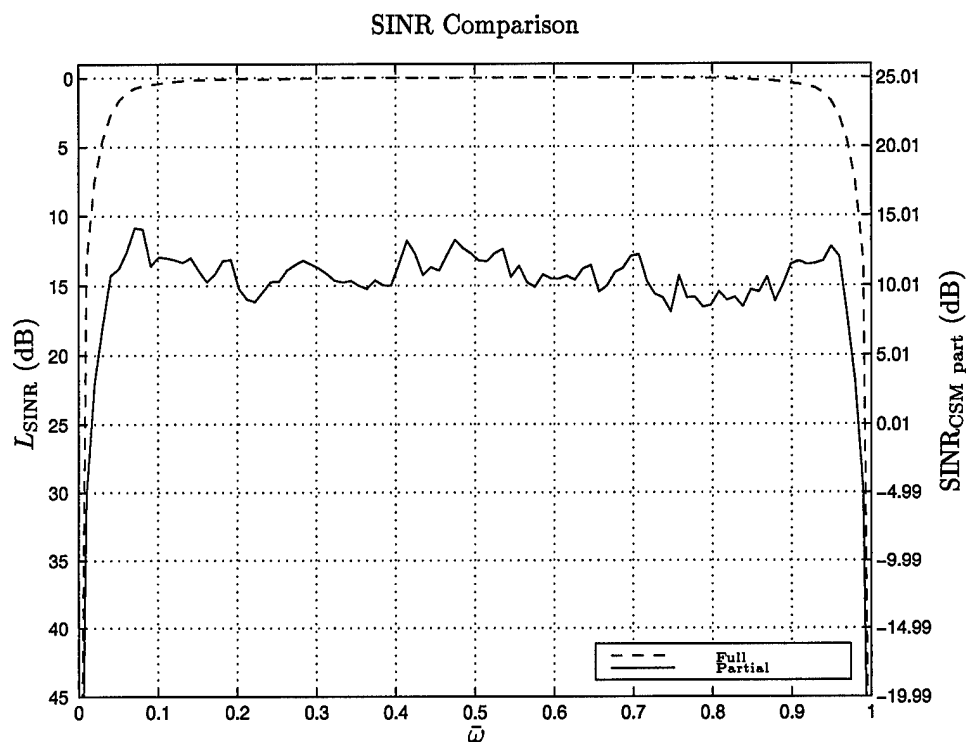


Figure 6.2 Reduced dimension SINR using the cross spectral metric compared to full dimensionality. The dimension is  $18 \times 18/16 \approx 20$ . The full dimensionality is  $18 \times 18 = 324$  DOF.

Figure 6.2 shows a plot of Eqn. (6.20) versus normalized Doppler  $\bar{\omega}$ . When compared to Fig. 3.20 for the FTS partially adaptive algorithm, the curve seems rather rough in texture. However, this behavior does make sense. The FTS algorithm is adaptive in the spatial domain only, hence the ripples in the curves shown in Fig. 3.20 versus  $\bar{\omega}$  represent the bins in the Doppler filter. The partially adaptive CSM model is adaptive in both the spatial and temporal domains. Hence, the DOF are chosen from both. This selection does not necessarily result in a smooth pattern when plotted versus  $\bar{\omega}$ . However, the partially adaptive CSM algorithm chooses the DOF according to the cross spectral metric, resulting in the optimum SINR as shown in Chapter IV.

At this point, it is interesting to compare the results to the those obtained in Chapter III for the FTS algorithm. Figure 6.2 corresponds to a partially adaptive algorithm with 20 DOF. Figure 3.20 corresponds to the FTS algorithm with 18 DOF. Comparing the two figures shows the

effect of a high quality Doppler filter on the FTS algorithm. The CSM algorithm is outperformed by FTS when either the Blackman Harris (-92 dB sidelobes, -6 dB/Octave roll-off) or the Hanning (-35 dB sidelobes, -18 dB/Octave roll-off) windows are used with the non-adaptive Doppler filter. The FTS algorithm performs better because the CSM algorithm simply cannot build the same quality filter with only 20 DOF. It must build a two dimensional filter whereas the FTS algorithm only has to construct a spatial dimension filter. Although the CSM algorithm doesn't produce the highest peak SINR, the curve is more nearly horizontal as a function of  $\bar{\omega}$ . Adding a spatial window as shown by Fig. 3.21 does improve the performance of the FTS algorithm in this regard, however the maximum SINR is correspondingly reduced. Using the spatial tapering, the Blackman Harris FTS algorithm is the only one to outperform the CSM algorithm.

### 6.3 Secondary Data Support for Covariance Estimation

As mentioned in Section 5.2, there is some question as to the validity of applying Reed's analysis of secondary data support to the CSM algorithm. Reed has shown the number of secondary data vectors required to be within 3 dB of the optimal SINR when using the Sample Matrix Inversion (SMI) technique is approximately twice the DOF or  $J = 2N_{\text{DOF}}$ .

The analysis done by Reed is beyond the scope of this thesis, therefore secondary data support requirements were determined experimentally. The first step in the analysis required the production of a known covariance matrix  $\mathbf{R}$ . For continuity in the thesis, the covariance matrix used in this analysis is the same as the one generated in Chapter III and used in all of the examples, with the exception of the MCARM (real aircraft data) examples found in Chapter V. This interference plus noise covariance matrix contains the effects of two jammers (located at  $-40^\circ$  and  $25^\circ$ ), clutter, and noise.

The data is generated using the standard technique for building colored data from zero mean unit variance Gaussian samples. The first step is generation of the base samples with the desired

Gaussian distribution,

$$\mathbf{x} \sim \mathcal{N}(0, 1), \quad (6.21)$$

where the notation shown is the standard for a zero mean unit variance Gaussian distribution and  $\mathbf{x}$  is a column vector of length  $MN$  (the length of the space-time snapshot).

If the desired covariance matrix (in our case, the one used in Chapter III) is denoted  $\mathbf{R}$ , the Cholesky decomposition  $\mathbf{Q}$  can be found such that

$$\mathbf{R} = \mathbf{Q}^H \mathbf{Q}. \quad (6.22)$$

Setting the space-time snapshot according to

$$\tilde{\chi} = \mathbf{Q}^H \mathbf{x}, \quad (6.23)$$

results in the data being colored according to  $\mathbf{R}$  as desired. The tilde above  $\tilde{\chi}$  denotes it is different than the real space-time snapshot  $\chi$ . Proof that the covariance matrix is indeed  $\mathbf{R}$  is easily shown.

If the covariance matrix of the generated samples is denoted  $\tilde{\mathbf{R}}$ , then

$$\begin{aligned} \tilde{\mathbf{R}} &= \mathcal{E} \{ \tilde{\chi} \tilde{\chi}^H \} \\ &= \mathcal{E} \{ \mathbf{Q}^H \mathbf{x} \mathbf{x}^H \mathbf{Q} \} \\ &= \mathbf{Q}^H \mathcal{E} \{ \mathbf{x} \mathbf{x}^H \} \mathbf{Q} \\ &= \mathbf{Q}^H \mathbf{I} \mathbf{Q} \\ &= \mathbf{Q}^H \mathbf{Q} \\ &= \mathbf{R}. \end{aligned} \quad (6.24)$$

From Chapter V and basic probability theory, we know the expected value in Eqn. (6.24) is simply the mean of an infinite number of samples. This gave rise to the Maximum Likelihood Estimator (MLE) for  $\hat{\mathbf{R}}$ ,

$$\hat{\mathbf{R}} = \frac{1}{J} \sum_{i=1}^J \mathbf{x}_i \mathbf{x}_i^H, \quad (6.25)$$

where  $J$  is the number of secondary data vectors used in the estimate. As  $J$  approaches infinity, the covariance matrix estimate  $\hat{\mathbf{R}}$  will approach  $\mathbf{R}$ . Using this relationship, we can perform a Monte Carlo simulation at each secondary data set size  $J$  and determine experimentally the required secondary data support to achieve a certain SINR.

As verification of the Monte Carlo simulation process, Fig. 6.3 is offered. The FTS algorithm is used with artificially generated space-time snapshots. This algorithm is adaptive in the spatial domain only. Since there are 18 antenna elements in this simulation, there are 18 DOF. Reed's rule predicts approximately  $2N_{\text{DOF}}$  secondary data vectors are required to achieve performance within 3 dB of optimal SINR.

In the experiment, there were 20 trials for each secondary data set size. The solid line depicts the experimental SINR Loss,  $L_{\text{SINR}}$ , for the secondary data set size shown on the  $x$ -axis and is the mean of the values returned from each of the 20 trials. The averaging technique used is the arithmetic mean, signifying the mean of the samples is calculated in the native units. The conversion to dB is done after the mean has been calculated. The standard deviation of these 20 trials is depicted by the dashed line. The standard deviation is also calculated in the native units and then converted to dB. The solid line in the figure reaches 3 dB of optimal with secondary data support between 30 and 35 vectors. As expected, these results coincide very well with Reed's prediction of  $2 \times 18 - 3 = 33$ .

Figure 6.4 shows a Monte Carlo simulation for the full dimension CSM algorithm. Again, the solid line depicts the average  $L_{\text{SINR}}$  for the 20 trials and the dashed line depicts the standard



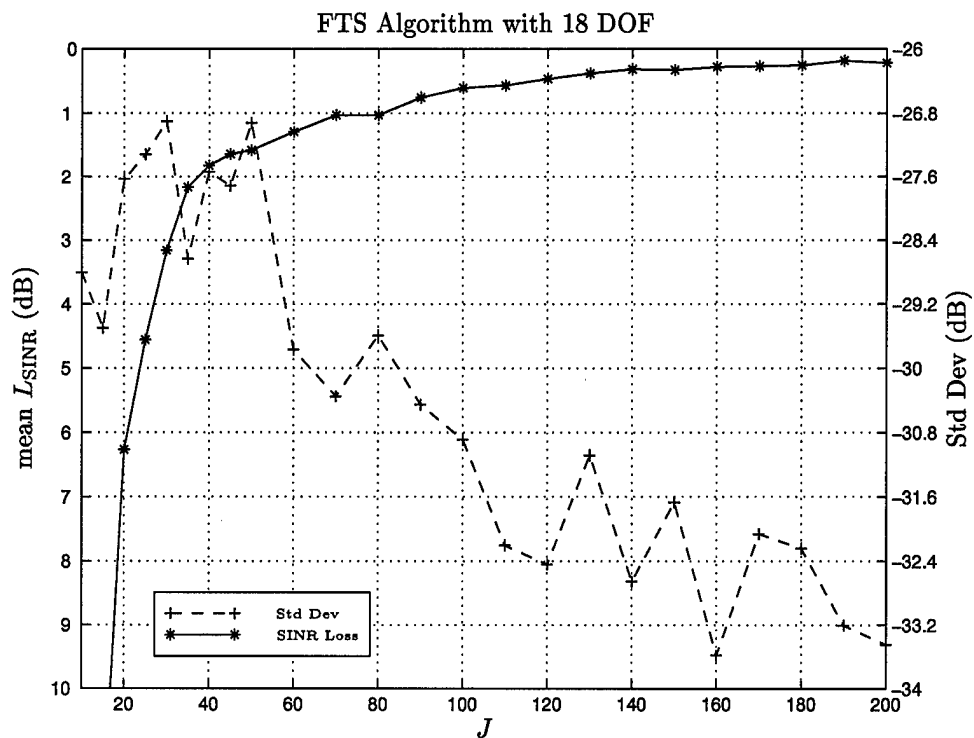


Figure 6.3 The experimental  $L_{\text{SINR}}$  curve versus secondary data set size for the FTS algorithm. There are 18 DOF, hence Reed predicts 33 secondary data vectors to achieve performance within 3 dB of optimal SINR. The Monte Carlo simulation indicates between 30 and 35 secondary data vectors are required, agreeing very well with Reed's prediction of 33. The analysis was done for Doppler bin 9 (of 18 bins), no windows/tapers were used.

deviation. Before examining the results shown in the figure, an explanation of the third curve is in order. This curve is characterized by the dash dot sequence and depicts the ratio of the mean SINR to the standard deviation of these experiments. When using the dB values, this ratio simply becomes the standard deviation subtracted from the mean. The ratio provides an indicator of the quality of the Monte Carlo simulation.

Typically, a ratio of 8 to 10 dB signifies enough trials were performed in the simulation. Higher ratios correspond to higher confidence in the results. According to Fig. 6.4, the ratio indicates the 20 trials used in the Monte Carlo simulation were adequate for this experiment. When the ratio is smaller as in Fig. 6.7, the  $L_{\text{SINR}}$  curve is not as smooth. This smoothness is another indication of the quality of the experiment. Large differences in mean  $L_{\text{SINR}}$  as the sample support is increased suggest not enough trials were done in the experiment. Although Fig. 6.7 exhibits a small ratio, the mean  $L_{\text{SINR}}$  values do not exhibit exaggerated differences as the secondary data support is increased. Therefore, the number of trials was not increased for the experiments. The ratio curve was omitted from Fig. 6.3 because the values were on the order of +30 dB, well above the scale used for the standard deviation curve making it undesirable to add this third curve to the plot.

Continuing the analysis of Fig. 6.4, the optimal SINR (known covariance) for full dimensionality was shown in Fig. 6.1 as 25 dB. The marked point in Fig. 6.4 represents 3 dB within this optimal SINR of 25 dB. As expected for the full dimension model, this point corresponds to a data set size of approximately twice the DOF, or  $2 \times 324 = 648$ . Therefore, approximately 650 secondary data vectors are needed to estimate the covariance matrix and retain performance within 3 dB of the optimal SINR. The Monte Carlo simulation results in the conclusion that Reed's rule holds for the CSM algorithm when using full dimensionality.

The next step in the analysis is Monte Carlo experiments for reduced dimension CSM algorithms. However, in order to calculate  $L_{\text{SINR}}$  we must first determine the optimum SINR (known covariance) for the reduced dimensions of interest. This calculation is required because the opti-

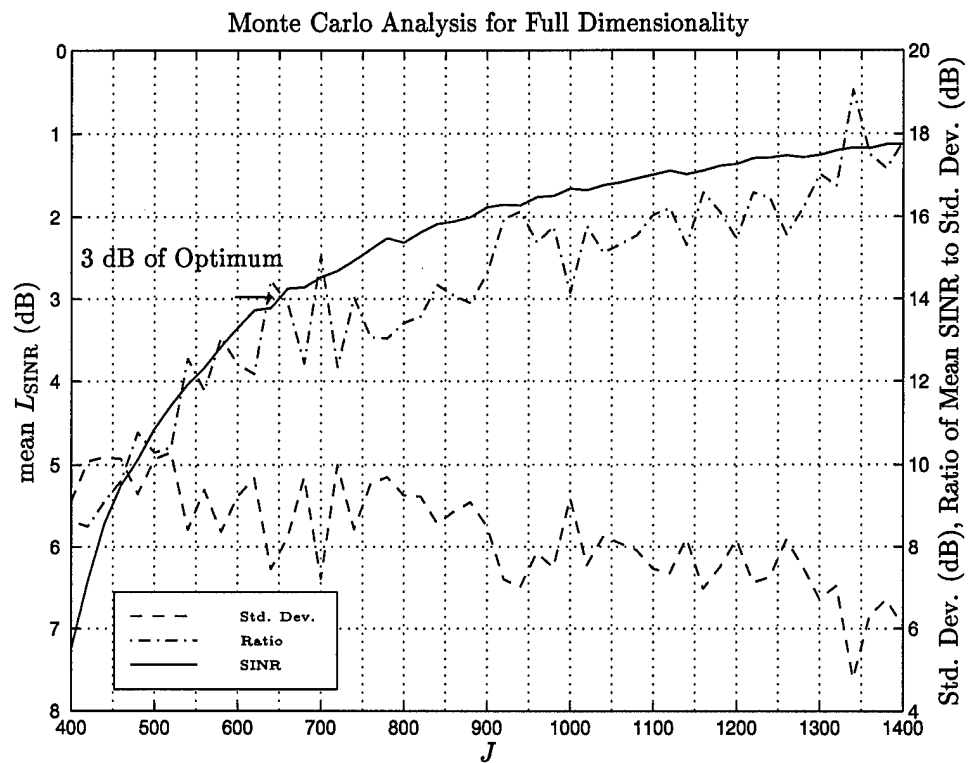


Figure 6.4 The  $L_{\text{SINR}}$  curve versus secondary data set size for the full dimensioned CSM model. As expected for full dimensionality, it is approximately twice the DOF ( $2 \times 324 = 648$ ).

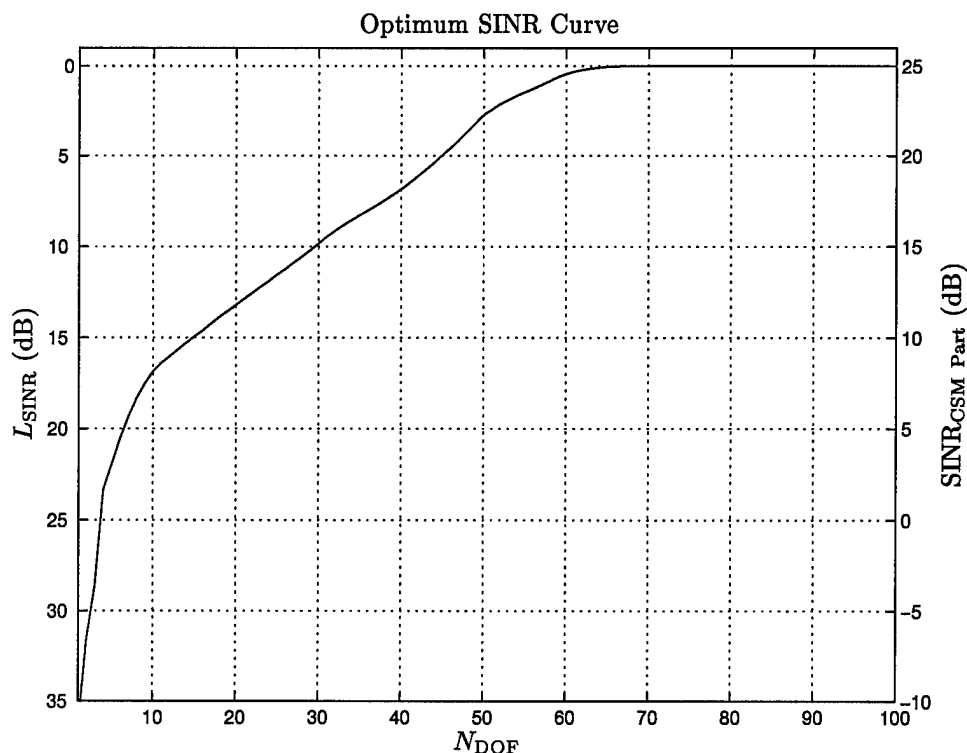


Figure 6.5 The optimum SINR curve (known covariance) versus weight vector dimension ( $N_{DOF}$ ). The SINR curve illustrates as few as 65 DOF are required to reach the maximum optimal SINR.

imum SINR will experience roll-off as the dimension of the problem is reduced. Figure 6.5 presents the optimum SINR curve.

Figure 6.5 is the reference needed to determine the  $L_{SINR}$  resulting from using the interference plus noise covariance matrix estimate. The figure also gives insight into the reason behind reduced dimension STAP algorithms. The optimum SINR curve rapidly approaches the maximum SINR of 25 dB with far fewer than  $MN$  DOF. In this example, the same interference plus noise covariance matrix is being used as in all previous examples. This covariance matrix corresponds to  $18 \times 18$ , or 324 DOF. The  $N_{DOF}$  required to reach the maximum optimum SINR is approximately 65, much less than the full dimension of 324.

Figure 6.6 is a plot of the experimental  $L_{SINR}$  versus secondary data set size  $J$  for the partially adaptive CSM algorithm using 65 DOF. The solid line indicates the mean  $L_{SINR}$  for the 20 trials

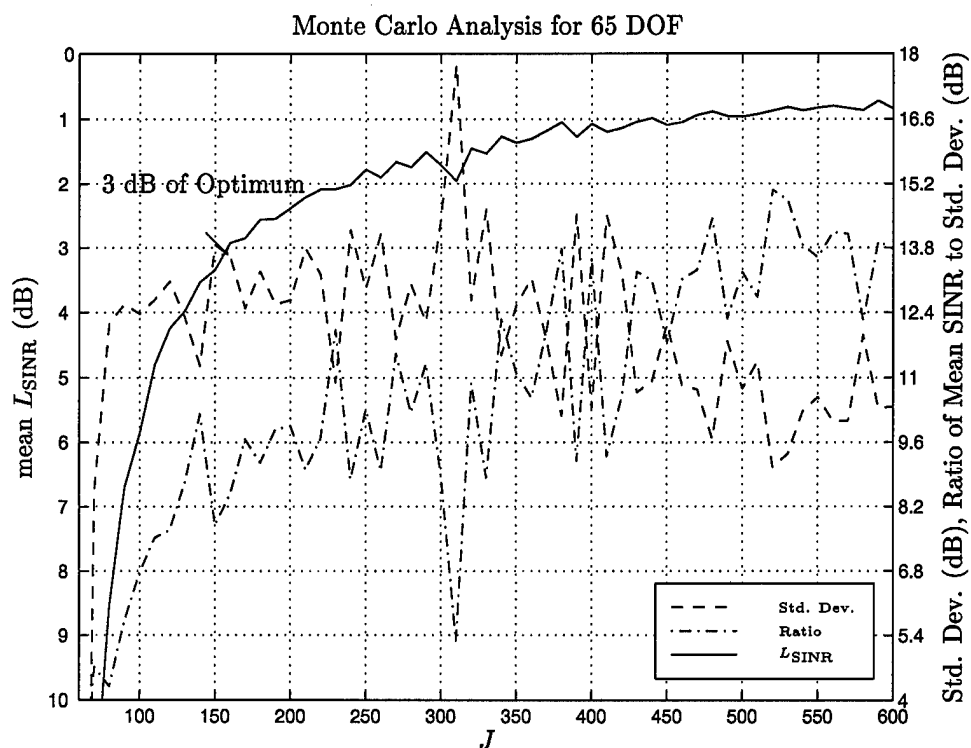


Figure 6.6 The  $L_{\text{SINR}}$  curve versus secondary data set size for the partial dimension CSM algorithm using 65 DOF. The labeled point represents 3 dB of optimal SINR (25 dB). Contrary to Reed's analysis, the number of secondary data vectors required is greater than twice the DOF ( $2 \times 65 = 130$ ). This reduced dimension model requires 160 secondary data vectors to achieve 3 dB of optimal performance.

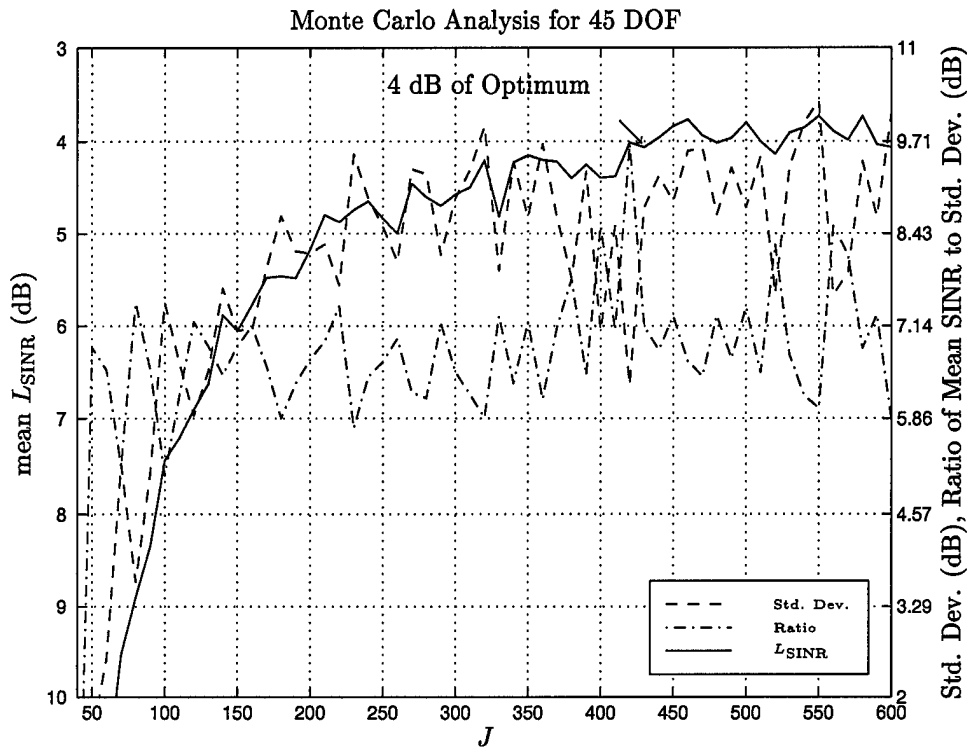


Figure 6.7 The  $L_{\text{SINR}}$  curve versus secondary data set size for the partial dimension CSM algorithm using 45 DOF. The labeled point represents 4 dB of optimal SINR (20 dB). Contrary to Reed's analysis, the curve does not reach 3 dB of optimal. This reduced dimension model requires 420 secondary data vectors or 9 times the DOF to achieve 4 dB of optimal.

at each secondary data set size. The dashed line corresponds to the standard deviation of these 20 trials. The data set size was increased in steps of 10. As before, the dash-dot line corresponds to the ratio of the mean SINR to the standard deviation. The ratio curve shown in the figure indicates the trial size of 20 was adequate for the simulation.

The results of this Monte Carlo simulation indicate 160 secondary data vectors are required to achieve SINR performance within 3 dB of optimal. This performance differs from Reed's rule stating the secondary data support should be approximately twice the DOF. Using the  $2N_{\text{DOF}}$  rule implies only  $2 \times 65 = 130$  secondary data vectors are needed, resulting in the loss of approximately 1 dB in SINR. The Monte Carlo simulation indicates approximately 2.5 secondary data vectors are required for operation within 3 dB of optimal.

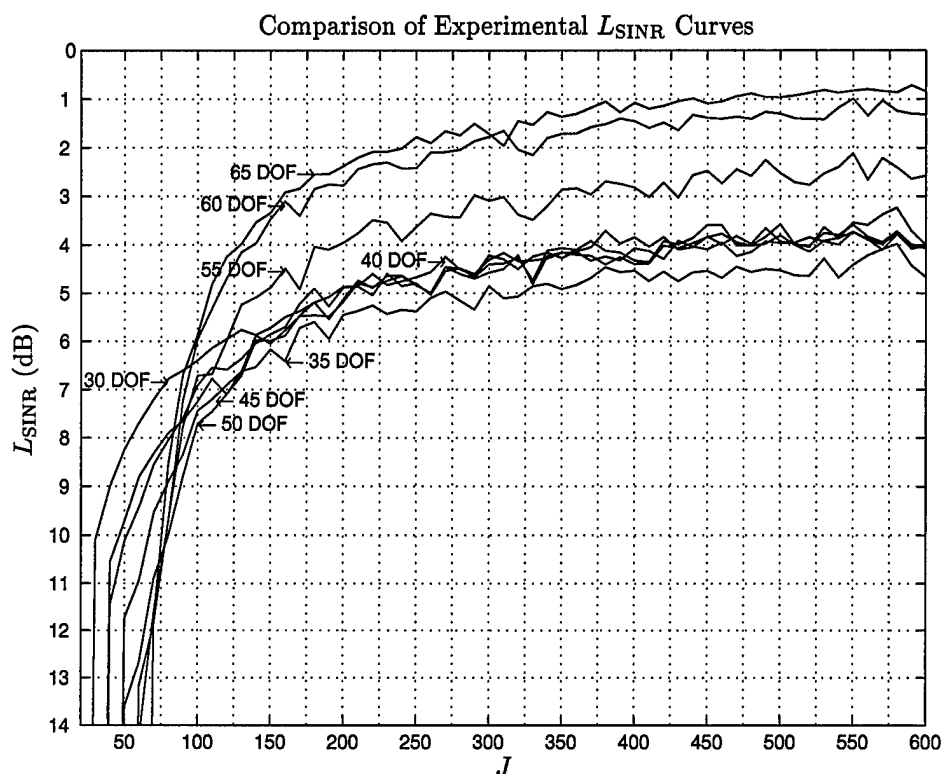


Figure 6.8 Overlay of the  $L_{\text{SINR}}$  curves for reduced dimensions below the dimension of the interference subspace. Performance relative to optimal is reduced as the DOF are reduced.

The next simulation reduces the DOF to 45. Fig. 6.5 shows the optimum SINR (known covariance) is now 20 dB. Therefore, the curve needs to converge to 17 dB. The Monte Carlo simulation in Fig. 6.7 shows the  $L_{\text{SINR}}$  curve never reaches this point. To achieve within 4 dB of optimal SINR requires secondary data support of size 420. This translates to  $420/45 \approx 9$  times the DOF!

Overlaying Monte Carlo simulation results for several different reduced dimensionalities shows the trend of interest. Figure 6.8 is a plot the experimental SINR Loss ( $L_{\text{SINR}}$ ) curves for reduced dimension applications with the DOF below the dimension of the interference subspace. Each curve has been referenced to the appropriate optimum SINR values from Fig. 6.5. The dimension of the interference subspace was determined through the use of two tools, the Singular Value

Decomposition (SVD) of the interference plus noise covariance matrix  $\mathbf{R}$  and plots of the Cross Spectral Metric (CSM).

A note about the curves in Fig. 6.8 is in order. The sharp drop-off each curve exhibits corresponds to the point where  $J$  is less than  $N_{\text{DOF}}$ . This drop-off occurs because the number of secondary data vectors used in the interference plus noise covariance matrix estimate  $\hat{\mathbf{R}}$  must be at least equal  $N_{\text{DOF}}$ . This requirement is a result of the CSM algorithm using an eigenvalue decomposition to reduce the dimensionality of the problem. If only  $J$  secondary data vectors are used to estimate  $\hat{\mathbf{R}}$  and  $J < N_{\text{DOF}}$ , then only the first  $J$  eigenvalues are nonzero. Referencing Eqn. (6.19), we see the inverse of  $\mathbf{\Lambda}_O$  (the diagonal matrix of eigenvalues corresponding to the eigenvectors chosen by the CSM) is required. If there are only  $J$  nonzero eigenvalues, then  $\mathbf{\Lambda}_O$  contains eigenvalues that are either equal to zero or very small. Therefore, the matrix is very close to singular and cannot be inverted, resulting in the sharp drop-off reflected in each  $L_{\text{SINR}}$  curve shown in the graphs.

Figure 6.9 shows the magnitude of the singular values and the CSM. The upper ledge represents the dimension of the interference subspace and the lower ledge is the noise (uncorrelated/white). This plot reinforces the fact that it is the noise that makes  $\mathbf{R}$  invertible. Without it, the rank of  $\mathbf{R}$  is equal to the dimension of the interference subspace, which is less than  $MN$ .

Figure 6.10 is simply an expanded view of Fig. 6.9. This allows a more accurate estimation of the dimension of the interference subspace. From the figure, the dimension is approximately 70.

These two figures add considerable insight into the underlying architecture of the CSM algorithm. The model chooses the DOF that cancel the interference, not the noise, taking advantage of the statistical properties of the incoming data. Combining the knowledge gained from Figs. 6.9 and 6.10 with Fig. 6.8 results in the prediction that  $N_{\text{DOF}}$  should be equal to the dimension of the interference subspace for optimum performance in situations of unknown covariance, although



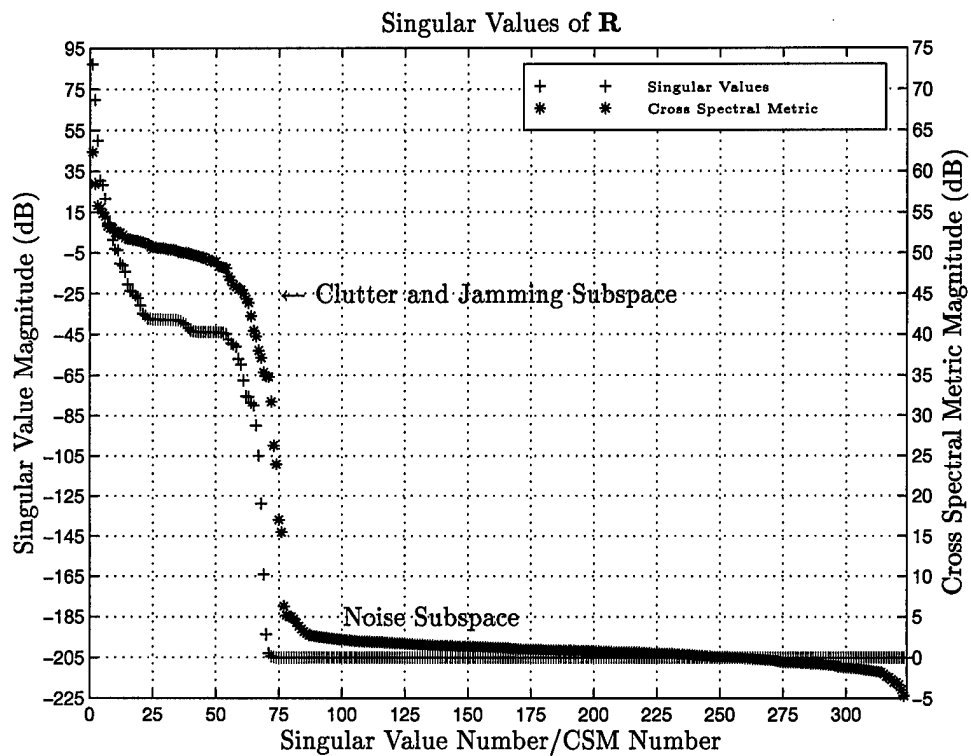


Figure 6.9 Singular value and cross spectral metric magnitudes of the known interference plus noise covariance matrix. The singular values represent the power contained in the interference subspace. This figure illustrates the most power is contained in the clutter and jamming interference subspace. The noise subspace represents very little of the overall power.

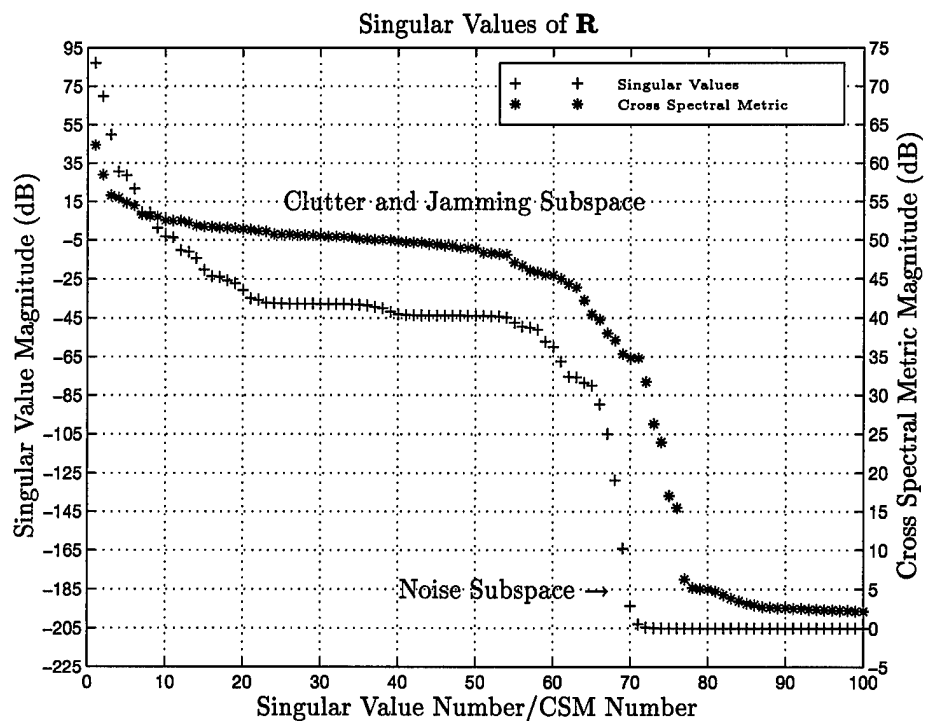


Figure 6.10 Closer examination of the most significant singular values indicates the most power is contained in the first 70, a measure of the dimension of the interference subspace. This plot coincides with Fig. 6.5 in illustrating the DOF required to reach optimum is equal to the dimension of the interference subspace.

obviously determining the dimension of the interference subspace for real time operation is very expensive.

Going back to the Monte Carlo analysis, Fig. 6.8 also shows an unexpected trend in the data. This trend is in direct conflict to Reed's rule. As  $N_{\text{DOF}}$  is reduced, the required secondary data set size  $J$  to achieve a constant measure of the optimum performance *increases*. For example, performance within 3 dB of optimal is obtained with 160 secondary data vectors for 65 DOF while 175 are required when the  $N_{\text{DOF}}$  is reduced to 60. Further reduction of  $N_{\text{DOF}}$  results in SINR performance that does not reach 3 dB of optimal. Each experimental  $L_{\text{SINR}}$  curve is referenced to the optimum SINR corresponding to  $N_{\text{DOF}}$ .

The figure also shows a distinct thresholding effect occurring when  $N_{\text{DOF}}$  is decreased to values below 55. These curves are grouped together with all producing performance within 5 or 6 dB of optimal using approximately the same number of secondary data vectors. These results contradict the primary goal of reducing the DOF to gain a corresponding decrease in secondary data support.

Figure 6.11 overlays the experimental  $L_{\text{SINR}}$  curves for DOF above the dimension of the interference subspace. Again, the curves illustrate the sharp drop-off in performance when  $J$  is decreased below  $N_{\text{DOF}}$ . Contrary to the overlay shown in Fig. 6.8, increasing the DOF does not increase the performance of the algorithm. Performance is significantly *decreased* when  $J$  is equal to  $MN$ , or 324. Maximum performance is again obtained when  $N_{\text{DOF}}$  is equal to the dimension of the interference subspace. Note that once  $J$  is increased beyond  $MN$ , the curves converge to the same point.

**6.3.1 Hypothesis Behind Poor Performance.** The Monte Carlo simulation has shown the performance of the CSM algorithm when estimating the covariance is maximized when  $N_{\text{DOF}}$  is equal to the dimension of the interference subspace. Both Figs. 6.8 and 6.11 support this statement. The poor performance of the algorithm when  $N_{\text{DOF}}$  is not equal to the dimension of the interference

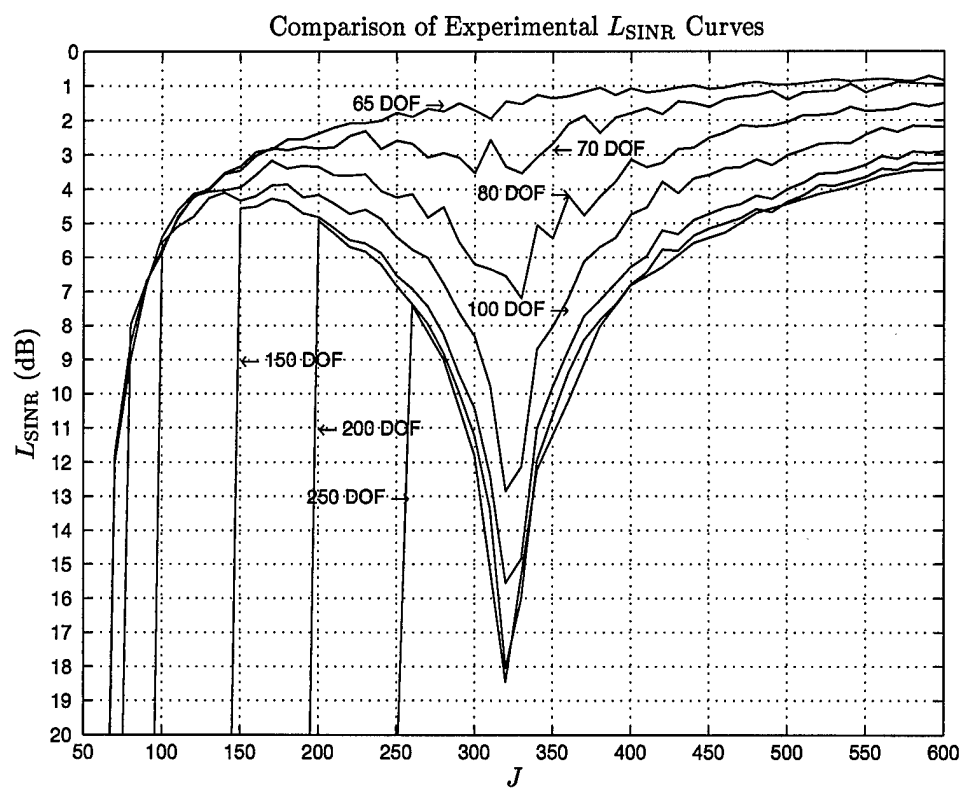


Figure 6.11 Overlay of the experimental SINR curves for reduced dimensions above the dimension of the interference subspace. Performance relative to optimal is reduced as the DOF are increased.

subspace is likely due to calculation noise in the eigenvalues of  $\hat{\mathbf{R}}$ . Small secondary data set sizes result in an estimated interference plus noise covariance matrix  $\hat{\mathbf{R}}$  of less than full rank. In the examples used, this statement is true for any secondary data set size less than  $18 \times 18$ , or 324.

As corroborated by Fig. 6.9, the first 65 to 70 singular values are much larger than any subsequent singular values. The singular values are directly related to the eigenvalues [20], hence Fig. 6.9 gives appropriate insight. The largest eigenvalues are not affected as greatly by the varying secondary data set sizes. However, the smaller eigenvalues are greatly affected.

Referencing Eqn. (4.71), the CSM is defined as

$$\left| \frac{\nu_i \mathbf{r}_{\mathbf{b}d}}{\sqrt{\lambda_i}} \right|^2, \quad (6.26)$$

where  $\mathbf{r}_{\mathbf{b}d}$  is the cross correlation between the vector  $\mathbf{b}$  and the desired scalar  $d$ ,  $\nu_i$  is the  $i^{th}$  eigenvector, and  $\lambda_i$  is the corresponding eigenvalue. The eigenvalue error will be small relative to the magnitude of the largest eigenvalues, however the error is large relative to the small eigenvalues.

This error in the small eigenvalues results in an inherent instability in the CSM algorithm when  $N_{\text{DOF}}$  is increased to include the DOF corresponding to small eigenvalues. The dimension of the interference subspace marks the point at which the eigenvalues become much smaller. A small change in secondary data set size can result in a drastically different  $\mathbf{U}$ , the dimension reducing matrix of eigenvectors. Remember, the eigenvectors comprising  $\mathbf{U}$  are chosen according to the CSM.

In the examples shown, the dimension of the interference subspace was approximately 65. This is exactly the point at which the performance of the CSM algorithm diminishes, as shown by Fig. 6.11. Once the secondary data set size  $J$  is increased beyond  $MN$  (324 in the examples) the performance begins to improve again. This is explained through an examination of the reduced dimension weight vector shown in Eqn. (6.19).

Let's continue the analysis by examining the performance of the  $L_{\text{SINR}}$  curves in a manner from left to right or increasing  $J$ . When  $J < MN$ , there are only  $J$  nonzero eigenvalues because  $\hat{\mathbf{R}}$  (of dimension  $MN \times MN$ ) is not full rank. As long as  $N_{\text{DOF}} \leq J$ ,  $\mathbf{\Lambda}_O$  contains only large eigenvalues. The matrix is easily inverted. However, increasing  $N_{\text{DOF}}$  beyond  $J$  forces  $\mathbf{\Lambda}_O$  to contain eigenvalues that are very close to zero. Inverting this matrix is difficult and prone to errors. This is shown by the sharp drop-off each  $L_{\text{SINR}}$  curve exhibits when  $J < N_{\text{DOF}}$ .

When  $N_{\text{DOF}}$  is equal to the dimension of the interference subspace, the  $L_{\text{SINR}}$  performance is maximum. This is explained by again looking at the eigenvalues. If  $\hat{\mathbf{R}}$  was full rank, the eigenvalue decomposition would contain  $MN$  nonzero eigenvalues. However, since  $J$  is still less than  $MN$ , the matrix is not full rank and contains only  $J$  nonzero eigenvalues. The rest are either zero or very close to zero depending on the current random realization of  $\hat{\mathbf{R}}$ . These small eigenvalues create calculation noise in the CSM. When  $N_{\text{DOF}} = J$ , the CSM algorithm is not forced to choose eigenvectors lying in the noise subspace and it gets to use all of the eigenvectors lying in the interference subspace. Hence, the performance of the algorithm is the best.

Choosing  $J < MN$  and increasing  $N_{\text{DOF}}$  such that it is larger than the dimension of the interference subspace forces the CSM algorithm to choose eigenvectors lying in the noise subspace. These eigenvectors are characterized by the extremely small eigenvalues. These eigenvalues then become a part of  $\mathbf{\Lambda}_O$ . When this matrix is inverted, the small eigenvalues result in the interference estimate being larger than it should be and a corresponding increase in  $L_{\text{SINR}}$ . The  $L_{\text{SINR}}$  increases as  $N_{\text{DOF}}$  increases because more of the small eigenvalues are becoming part of  $\mathbf{\Lambda}_O$  and corrupting the interference estimate. This is shown by the results in Fig. 6.11.

Once  $J$  has reached  $MN$ , the interference plus noise covariance matrix estimate  $\hat{\mathbf{R}}$  has reached full rank and none of the eigenvalues are zero, although errors still exist because of the small secondary data set size. The  $L_{\text{SINR}}$  curves begin the expected progression to optimal performance as  $J$  is increased.

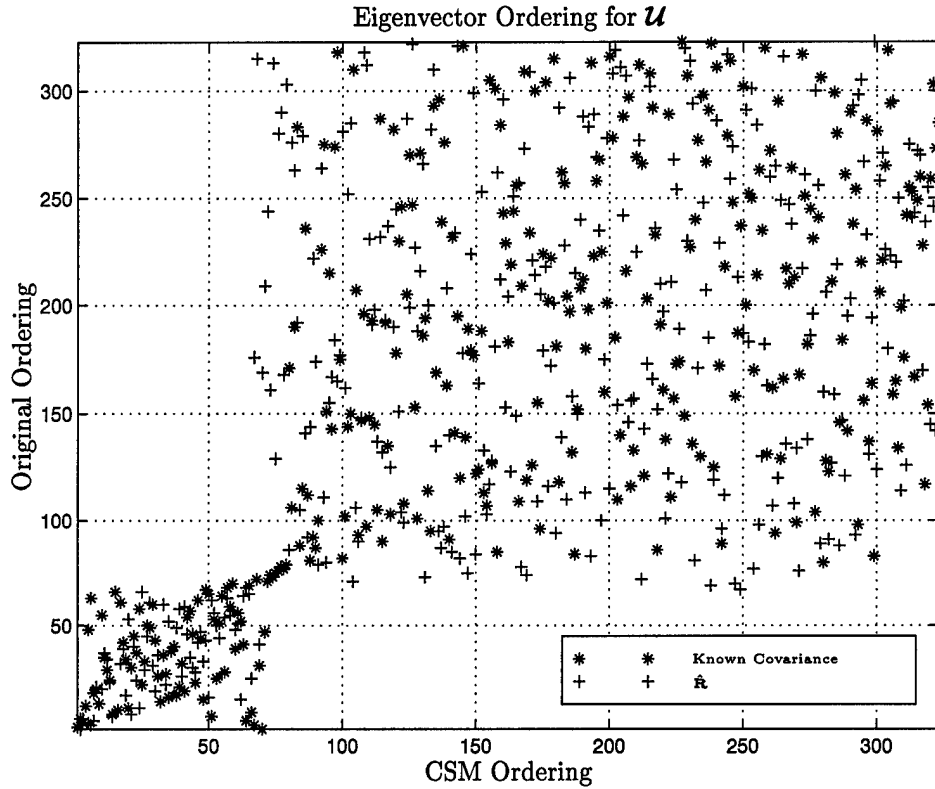


Figure 6.12 The CSM ordering of the eigenvectors for a secondary data set size  $J$  equal to 323, one less than  $MN$ . The ordering is significantly different for estimated covariance,  $\hat{\mathbf{R}}$ . Changing  $J$  results in different ordering. The eigenvectors in the lower left corner lie in the interference subspace.

There is still some question as to why the performance of the CSM algorithm experienced decreasing performance with  $N_{\text{DOF}}$  below the dimension of the interference subspace. Figure 6.12 gives insight into this problem. The figure represents the order in which the DOF are chosen by the CSM algorithm for two different scenarios.

The first scenario assumes a known interference plus noise covariance matrix,  $\mathbf{R}$ . These points can be considered the optimal ordering. The  $x$ -axis is the order in which the eigenvectors are chosen and the  $y$ -axis is the original order they were in. For example, the first eigenvector or DOF in the optimal ordering was originally the second one. It's location on the plot is position (1,2). This means the second DOF has the greatest positive effect (according to the CSM) on the SINR than any of the other DOF.

The second scenario in the plot is for an estimated interference plus noise covariance matrix  $\hat{\mathbf{R}}$ . The data was generated in the same manner as in the Monte Carlo experiments, using the coloring process explained in that section. Only 323 secondary data vectors were used in this particular estimate.

As Fig. 6.12 shows, the ordering when using  $\hat{\mathbf{R}}$  is drastically different. The first 65 eigenvectors are the same as indicated by the clustering in the lower left corner of the plot. However, the order they are chosen has changed. This cluster contains the eigenvectors that lie in the interference subspace. As long as  $N_{\text{DOF}}$  is greater than 65, the ordering of these first data vectors doesn't matter because the algorithm is using all the vectors lying in the interference subspace. Decreasing  $N_{\text{DOF}}$  below 65 places emphasis on the ordering. This explains the decreased performance when  $N_{\text{DOF}}$  is below the dimension of the interference subspace, shown by Fig. 6.8.

#### 6.4 Non-Homogeneities and the CSM Algorithm

In Chapter V, the impact of non-homogeneities in the secondary data on the output of the Factored Time-Space (FTS) algorithm was explored in detail. The Generalized Inner Product (GIP) [14] was introduced in an effort to alleviate this impact. The GIP was shown, using MCARM data containing actual airborne radar measurements, to result in significant improvement in the output of the FTS algorithm. This improvement consisted of increasing the separation between the target peak and largest noise level. The chapter also explored the mathematical architecture behind the GIP non-homogeneity detector.

In this chapter, an analysis of the CSM algorithm is being presented. The next logical step in this analysis is an examination of the effects of non-homogeneities in the secondary data on the CSM algorithm.

The examination begins exactly as the Monte Carlo simulation for required secondary data support began. A Cholesky decomposition of the desired interference plus noise covariance matrix



$\mathbf{R}$  gives a method to color artificially generated white data. This method is explained in Section 6.3. The same  $\mathbf{R}$  is used as in all previous examples.

The simulation begins by generating one random realization of a CPI datacube through the coloring process explained in Section 6.3. Using this one realization for the three different cases of interest allows a direct comparison of the results. The three cases of interest include operating the CSM algorithm on data with only a target, data with a target and one non-homogeneity, and the CSM algorithm with GIP non-homogeneity detection on data with a target and one non-homogeneity. The purpose of this section is not to generate a Monte Carlo simulation. The analysis only requires generation of one set of data to determine the effects of non-homogeneities.

With this goal in mind, a single target was placed in the CPI datacube. This target was placed at range cell (1) 100. To simulate a non-homogeneity in the secondary data, a second target was placed at range cell 50. This second target is identical to the first, with the exception of the range cell location.

This non-homogeneity is a worst case scenario for the CSM algorithm. It represents worst case because the non-homogeneity is the same angle and Doppler as the desired target/steering vector. The CSM algorithm cannot filter any of it out. The gains seen here are due solely to the use of the GIP detection scheme.

Figure 6.13 illustrates the output of the CSM algorithm both with and without the non-homogeneity in the data. The interference plus noise covariance matrix estimate  $\hat{\mathbf{R}}$  was found using a symmetric window of secondary data vectors about the range cell of interest. The CSM algorithm was programmed to use 45 DOF and 200 secondary data vectors. The choice of 200 secondary data vectors was made by referencing Fig. 6.8. This choice of secondary data support should give performance within approximately 5 dB of optimal according to the Monte Carlo simulations. Of course, choosing 65 DOF results in better performance (within  $\approx 2$  dB of optimal SINR).

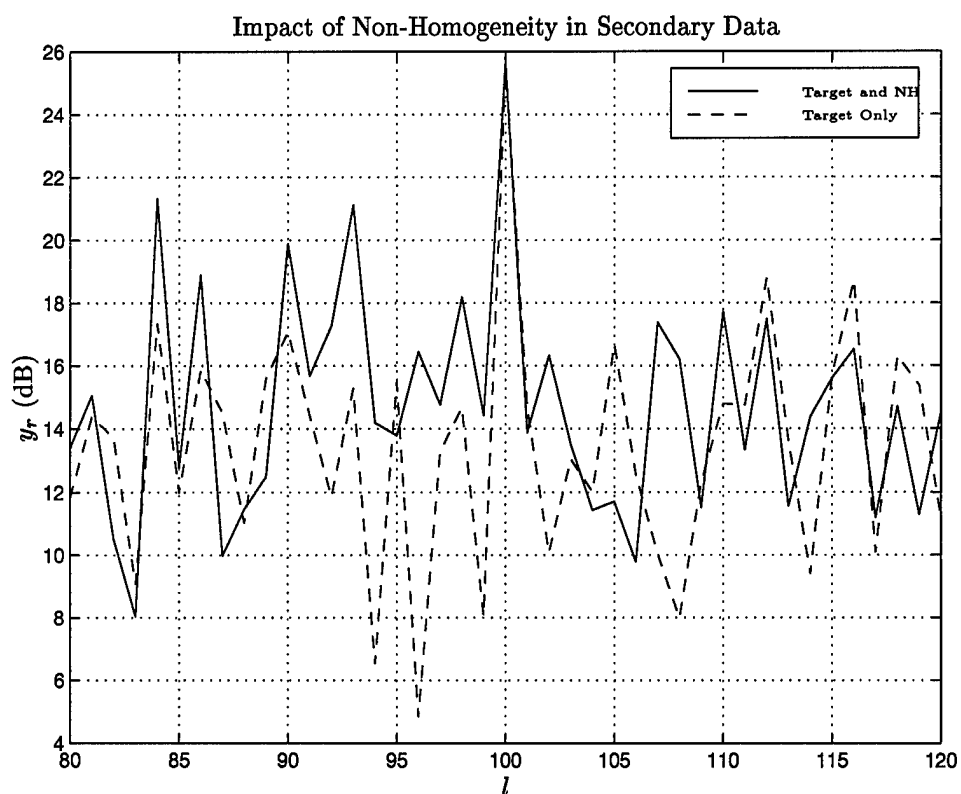


Figure 6.13 The impact of non-homogeneities in the secondary data on the CSM algorithm. This example shows the results of inserting a target in range cell 50 identical to desired target at range cell 100. The CSM algorithm utilizes 45 DOF and  $\hat{\mathbf{R}}$  is built with 200 symmetric secondary data vectors.

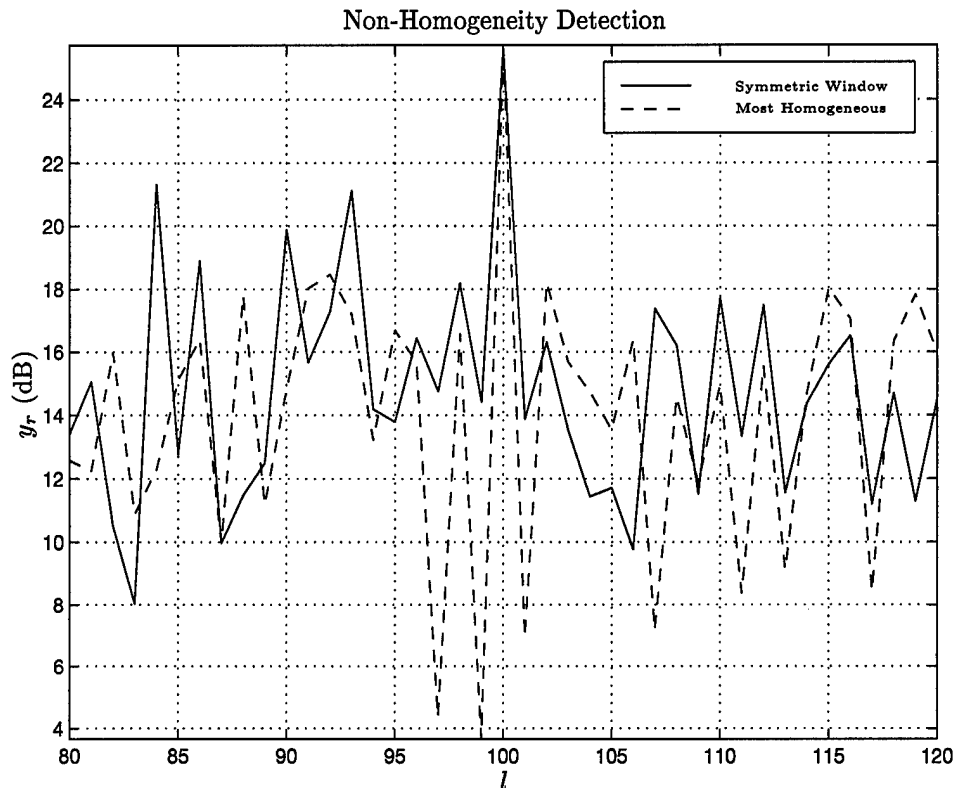


Figure 6.14 The impact of using the GIP non-homogeneity detector with the CSM algorithm. The curve is the output of the CSM algorithm using 45 DOF and 200 secondary data vectors.

The output of the CSM algorithm clearly shows the target at range cell 100. The non-homogeneity has not been inserted into the data resulting in acceptable performance when using a symmetric window of secondary data vectors. However, introducing the non-homogeneity degrades the performance of the algorithm. This is shown by the second curve in the graph. The separation from target peak to next highest peak has been reduced by approximately 3 dB. Since this output is typically threshold detected, this is a significant decrease.

Figure 6.14 illustrates the positive impact the GIP non-homogeneity detector has on the CSM algorithm. The separation between target and next highest peak has been increased from 5 dB to 8 dB. This is an improvement of 60%. Simply out of curiosity, the experiment was repeated after modifying the CSM algorithm to make use of 65 DOF.

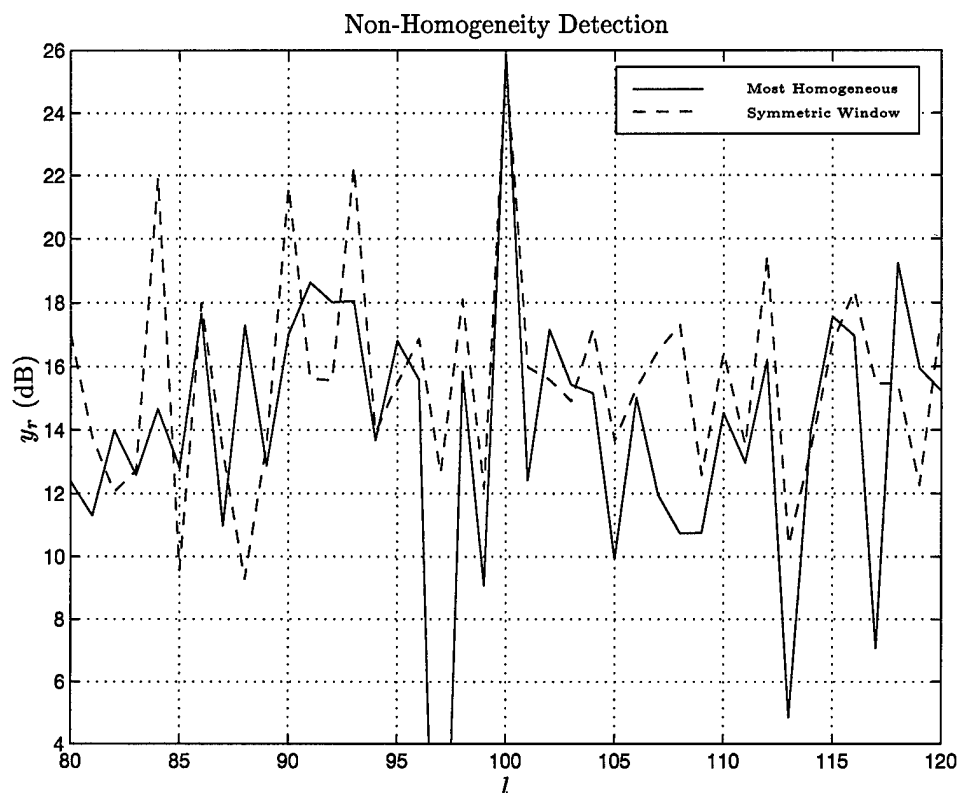


Figure 6.15 The impact of using the GIP Non-Homogeneity detector with the CSM algorithm. The curve is the output of the CSM algorithm using 65 DOF and 200 secondary data vectors.

Figure 6.15 shows the results of this second experiment. Clearly, the GIP non-homogeneity detector is a significant improvement. The separation between target and next highest peak has increased from 4 dB to 7 dB. An increase of 75%.

There is one major limitation of the GIP. It requires the inverse of a covariance matrix estimate based on the full secondary data set,  $\hat{\mathbf{R}}_s$ . To do the inverse,  $\hat{\mathbf{R}}_s$  must be full rank. This is a limiting factor when using the GIP. The CSM algorithm requires the estimation of a full dimensioned  $\mathbf{R}$ ,  $MN \times MN$ . Hence,  $\hat{\mathbf{R}}_s$  is also  $MN \times MN$ . This translates to a secondary data support requirement of at least  $MN$  vectors for implementation of the GIP detection scheme. This is in direct conflict to the primary goal of reducing secondary data support. For the experiments in this section, the secondary data set size was increased to 324 data vectors to ensure the invertibility

of  $\mathbf{R}_s$ . Symmetric windowing used 200 secondary data vectors symmetric about the range cell of interest while the GIP detection equipped algorithm used the 200 most homogeneous secondary data vectors.

### 6.5 Summary

The analysis performed in this chapter has uncovered several interesting pieces of information regarding the CSM algorithm. Expressions for the SINR were developed for both full and reduced dimension cases. Plots of these expressions using known covariance served to gain insight into the operation of the CSM algorithm. Similar plots (Chapter III) for the FTS algorithm allowed comparison between the algorithms.

As mentioned previously, the CSM algorithm requires the estimation of a full dimensioned interference plus noise covariance matrix. There is no gain over the original fully adaptive model in this respect. Furthermore, the eigenvalue/eigenvector decomposition of this matrix must be performed. Obviously, this algorithm does not enjoy the benefits of working in smaller dimensions such as FTS does. Calculation times are much greater for this algorithm than for FTS. For this reason, the CSM algorithm will likely remain a theoretical tool.

Although the CSM algorithm is computationally expensive, it does give optimum performance when the dimension reducing matrix is composed of eigenvectors. This was shown in Chapter IV. The point to remember is the dimension reducing matrix does not have to be composed of eigenvectors. There may be another matrix that gives better SINR performance.

The eigenvalue/eigenvector decomposition of  $\hat{\mathbf{R}}$  in the CSM algorithm brought into question the validity of Reed's rule. In an effort to answer this question of required sample support, a Monte Carlo analysis was performed. This analysis showed Reed's rule does hold when the CSM algorithm is fully adaptive. However, the rule does not hold when  $N_{\text{DOF}}$  is reduced below  $MN$  (fully adaptive).

The required secondary data support is related to the dimension of the interference subspace. The Monte Carlo analysis showed  $N_{\text{DOF}}$  should be equal to the dimension of the interference subspace for optimal SINR performance. For the interference plus noise covariance matrix used in this experiment, the number of required secondary data vectors should be set to approximately  $2.5N_{\text{DOF}}$ .

As a final point of interest, the GIP non-homogeneity detector was introduced into the CSM algorithm. A worst case situation was used in that the non-homogeneity inserted into the data was comprised of a duplicate target in another range cell. This represents worst case because the non-homogeneity does not lie in the interference subspace. The CSM algorithm cannot filter it out. The GIP non-homogeneity detector was shown to significantly improve the performance of the algorithm in this case.

The most significant problem with implementing the GIP non-homogeneity detector is the requirement of a fully ranked covariance estimate based on the entire secondary data set. This requirement conflicts with the goal of reducing the necessary secondary data support. Using the GIP detection scheme with the CSM algorithm requires a minimum of  $MN$  secondary data vectors.

## VII. Conclusions and Recommendations

### 7.1 Conclusions

One of the significant design considerations for any STAP algorithm is the required secondary data support. For an SMI algorithm, Reed found the required secondary data support as a function of the number of DOF [15] used in the adaptive algorithm. Through an exhaustive derivation based on the probability of detection  $P_D$ , Reed determined this support as approximately  $2N_{\text{DOF}}$ . A Monte Carlo simulation was used to experimentally determine the required secondary data support for the CSM algorithm.

It is interesting to relate Reed's rule to the physical size of the interference plus noise covariance matrix estimate  $\hat{\mathbf{R}}$ . For SMI techniques, this sample matrix is immediately inverted to obtain the adaptive weight vector. Hence, its dimension is  $N_{\text{DOF}} \times N_{\text{DOF}}$ . The number of secondary data vectors used to form the estimate must be at least  $N_{\text{DOF}}$  or the matrix is singular and cannot be inverted. Using  $2N_{\text{DOF}}$  secondary data vectors ensures that not only is the matrix invertible or non-singular but also that performance of the STAP algorithm is within 3 dB of optimal SINR.

The CSM algorithm uses the eigenvalue/eigenvector decomposition of a full dimensioned  $\hat{\mathbf{R}}$  to choose  $N_{\text{DOF}}$  out of a total of  $MN$  DOF. Correct determination of the eigenvalues and eigenvectors for the interference and noise subspaces require a full rank  $\hat{\mathbf{R}}$ . However, applying Reed's rule typically results in  $\hat{\mathbf{R}}$  being less than full rank. This introduces noise into the eigenvalue/eigenvector decomposition. This calculation noise compromises the ability of the algorithm to correctly determine the optimal DOF by corrupting the metric used to rank order the DOF.

The problem can be alleviated through a careful choice of  $N_{\text{DOF}}$ . Setting this value equal to the dimension of the interference subspace minimizes the impact of the calculation noise in the eigenvalue/eigenvector decomposition. The minimization occurs because the calculation errors in the eigenvalue/eigenvector decomposition have greater effect on the DOF corresponding to the noise subspace than those in the interference subspace. The greater effect is due to the smaller

eigenvalues associated with the eigenvectors spanning the noise subspace. Since their magnitude is smaller, the relative error is much larger. By choosing the DOF where the error has the least impact, the efficiency of the algorithm is optimized.

Decreasing  $N_{\text{DOF}}$  below the dimension of the interference subspace results in a corresponding decrease in SINR performance. This decrease in performance occurs as a result of the errors in these eigenvalues. Although the relative error is small when compared to the magnitude of the eigenvalue, it is enough to drastically change the order in which the DOF are chosen. Since  $N_{\text{DOF}}$  is now smaller than the dimension of the interference subspace, the order has become important. There is a corresponding decrease in performance.

Increasing  $N_{\text{DOF}}$  above the dimension of the interference subspace also results in a decrease in SINR performance. However, the algorithm does converge to optimal as secondary data set size increases. The most peculiar behavior occurs when the secondary data set size approaches  $MN$ , the full rank of  $\mathbf{R}$ . There is a large drop in SINR with a minimum occurring at a secondary data set size  $J$  of  $MN - 1$ . When  $J$  increases beyond  $MN$  the curve resumes the expected upward slope towards optimal SINR.

The main result of this work is to choose  $N_{\text{DOF}}$  equal to the dimension of the interference subspace. Obtaining performance within 3 dB of optimal SINR then requires  $2.5N_{\text{DOF}}$  secondary data vectors. Reed's rule of  $2N_{\text{DOF}}$  was found to only apply in the case of full dimensionality. The problem with this approach is obtaining an accurate estimate of the interference subspace dimension. An accurate estimate by use of the eigenvalues or singular values requires more secondary data vectors than that needed to reach the desired SINR performance. A method of determining interference subspace dimension by other means is needed. Brennan's rule should be considered for this application.

As a second goal, a brief analysis of non-homogeneities in the secondary data was performed. A worse case scenario for the CSM algorithm was used in that the non-homogeneity introduced into



the secondary data was identical to the desired target in angle and Doppler. The non-homogeneity only differed in range. This represents worst case because the blocking matrix  $\mathbf{B}$  used by the CSM algorithm does not allow this signal to become part of the interference plus noise estimate.

The CSM algorithm was found to be susceptible to the single non-homogeneity in the secondary data. The difference between the target peak and the next highest peak in the algorithm's output was reduced by approximately 3 dB. Introducing more non-homogeneities in the data would typically result in even less separation.

Through the use of the GIP non-homogeneity detector, the separation between target and next highest peak was increased by 3 dB. This effectively negates the impact of the non-homogeneity. The GIP works very well with the CSM algorithm in this situation.

## *7.2 Recommendations for Future Research*

As mentioned in Chapter IV, the dimension reducing projection matrix  $\mathbf{U}$  in the CSM algorithm does not have to be comprised of eigenvectors. The CSM algorithm provides maximum SINR only for the case when eigenvectors are used to construct this projection matrix.

An excellent opportunity for future research is development of a different projection matrix. The STAP formulation should be kept in the GSC architecture because it offers excellent visualization of the overall process. It also presents a good analogy to the Wiener filter. Avoiding the eigenvalue/eigenvector decomposition would result in several advantages.

If a constant projection matrix can be defined, the calculation costs of a decomposition can be avoided. This cost is a very important consideration if the algorithm were to be considered for real time implementation.

A second advantage to a constant projection matrix is secondary data support. Avoiding the decomposition of the full dimensioned covariance matrix would allow the algorithm to fall under the SMI umbrella. Hence, Reed's rule would apply.

The major drawback to a constant projection matrix is it is no longer adaptive. The same DOF will be chosen regardless of the interference environment the algorithm is placed in. Although the algorithm as a whole will remain adaptive due to the reliance on the interference plus noise covariance estimate, the DOF chosen will not result in optimal performance. For this reason, a constant projection matrix may not be feasible. However, it is important to consider that a constant projection matrix forming a basis for likely interference scenarios might produce acceptable performance, albeit sub-optimal performance. The gains in calculation cost and secondary data support from avoiding a decomposition might far outweigh the loss in performance.

A second area of future research involves examining Brennan's rule as a method to predict the interference subspace dimension. The current method of determining this dimension involves the use of either the eigenvalues or the singular values. These methods typically require a greater number of secondary data vectors than that required to meet the desired SINR performance. If Brennan's rule gives a sufficiently accurate estimate, the algorithm could be optimized without a penalty in secondary data support.

Another potential area for research is the GIP non-homogeneity detector. As mentioned previously, the major drawback to this method is the inverse of a covariance matrix estimate  $\hat{\mathbf{R}}_s$  based on the full secondary data set. Performing the inverse requires the estimate  $\hat{\mathbf{R}}_s$  be full rank. This full rank requirement then translates to a corresponding requirement of secondary data support. In the case of the CSM algorithm, choosing  $N_{\text{DOF}}$  equal to the dimension of the interference subspace means only  $2.5N_{\text{DOF}}$  secondary data vectors are required. However, because  $\hat{\mathbf{R}}_s$  is estimated from full dimensioned space-time snapshots there are at least  $MN$  secondary data vectors required to ensure it is invertible. Typically,  $2.5N_{\text{DOF}} < MN$ . A new method of detecting and excising non-homogeneities would alleviate this problem of secondary data support.

### Appendix A. Kronecker Product

The Kronecker Product is simple in concept. Given two matrices **A** and **B**, the Kronecker Product is defined in the following manner [9],

$$\mathbf{A} \otimes \mathbf{B} \equiv \begin{bmatrix} A_{11}\mathbf{B} & A_{12}\mathbf{B} & A_{13}\mathbf{B} & \cdots & A_{1M}\mathbf{B} \\ A_{21}\mathbf{B} & A_{22}\mathbf{B} & A_{23}\mathbf{B} & \cdots & A_{2M}\mathbf{B} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{N1}\mathbf{B} & A_{N2}\mathbf{B} & A_{N3}\mathbf{B} & \cdots & A_{NM}\mathbf{B} \end{bmatrix}. \quad (\text{A.1})$$

Therefore, given an  $N \times M$  matrix **A** and  $P \times Q$  matrix **B** results in an  $NP \times MQ$  matrix.

The following table of identities holds for Kronecker products.

Table A.1 Properties of Kronecker Products

- |    |  |
|----|--|
| 1. | $(\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C}$           |
| 2. | $(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$                |
| 3. | $\alpha(\mathbf{A} \otimes \mathbf{B}) = (\alpha\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (\alpha\mathbf{B})$  |
| 4. | $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$  |
| 5. | $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$   |
| 6. | $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{B} \otimes \mathbf{D})$ |
| 7. | $(\mathbf{A} \otimes \mathbf{B}) = (\mathbf{A} \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{B})$                       |

### *Appendix B. CSM Blocking Matrix Determination*

Two simple methods for computing the signal blocking matrix are the singular value decomposition algorithm and the QR decomposition algorithm [7]. The singular value decomposition of a matrix  $\mathbf{A}$  produces

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \quad (\text{B.1})$$

where the matrices  $\mathbf{U}$  and  $\mathbf{V}$  are unitary and  $\mathbf{\Sigma}$  is diagonal. The left and right singular vectors are contained in  $\mathbf{U}$  and  $\mathbf{V}$ , respectively. If the input matrix  $\mathbf{A}$  is of dimension  $P \times C$ , then  $\mathbf{U}$  is  $P \times P$ ,  $\mathbf{V}$  is  $C \times C$ , and  $\mathbf{\Sigma}$  is a diagonal  $P \times C$  matrix of the singular values. Note that regardless of whether the input matrix is real or complex, the nonzero singular values are always real and positive. An excellent discussion of the singular value decomposition is contained in [20].

Using these definitions, the singular value decomposition of the complex conjugate transposed steering vector (dimension  $1 \times NM$ ) is

$$\mathbf{s}^H = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \quad (\text{B.2})$$

where  $\mathbf{U}$  is  $1 \times 1$ ,  $\mathbf{\Sigma}$  is  $1 \times NM$ , and  $\mathbf{V}$  is  $NM \times NM$ . Taking the hermitian of both sides results in

$$\mathbf{s} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^H. \quad (\text{B.3})$$

If we choose the blocking matrix such that

$$\mathbf{B} = \mathbf{V}(:, 2 : NM)^H, \quad (\text{B.4})$$

where the notation used is MATLAB<sup>®</sup> notation for taking all the rows of the matrix and columns 2 through  $NM$ , then the product of the blocking matrix and steering vector is

$$\mathbf{B}\mathbf{s} = \mathbf{V}(:, 2 : NM)^H \mathbf{V} \mathbf{\Sigma} \mathbf{U}^H. \quad (\text{B.5})$$

Let's examine product of the reduced and full dimension right singular vector matrices. First express them in terms of the right singular vectors themselves,

$$\mathbf{V}(:, 2 : NM)^H \mathbf{V} = \begin{bmatrix} \mathbf{v}_2^H \\ \mathbf{v}_3^H \\ \vdots \\ \mathbf{v}_{NM}^H \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{NM} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{NM-1} \end{bmatrix}, \quad (\text{B.6})$$

where the last step takes advantage of the orthonormal columns of  $\mathbf{V}$ .

Since the input matrix to the singular value decomposition is actually a row vector, there will be only one singular value for the decomposition of  $\mathbf{s}^H$ , i.e.

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \end{bmatrix}^T. \quad (\text{B.7})$$

Recognizing this fact and substituting Eqn. (B.6) into Eqn. (B.5) results in

$$\mathbf{B}\mathbf{s} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{NM-1} \end{bmatrix} \mathbf{\Sigma} \mathbf{U}^H = \mathbf{0}, \quad (\text{B.8})$$

the desired result.

The second method mentioned by Goldstein is the QR decomposition. This decomposition produces

$$\mathbf{A} = \mathbf{Q}\mathbf{R}, \quad (\text{B.9})$$

where  $\mathbf{Q}$  is  $P \times P$  and  $\mathbf{R}$  is an upper triangular matrix of dimension  $P \times C$  given  $\mathbf{A}$  is  $P \times C$ . The elements of  $\mathbf{R}$  are given by [20]

$$r_{ij} = \mathbf{q}_i^H \mathbf{a}_j, \quad (\text{B.10})$$

where  $\mathbf{q}_i$  and  $\mathbf{a}_j$  are the columns of  $\mathbf{Q}$  and  $\mathbf{A}$ , respectively.

The matrix  $\mathbf{Q}$  is calculated using the Gram-Schmidt orthogonalization process, hence the columns of  $\mathbf{Q}$  are orthonormal and the matrix is termed orthogonal. The orthogonalization process produces a set of vectors that when taken as a whole represent the subspace spanned by the input matrix.

The QR decomposition of the steering vector is

$$\mathbf{s} = \mathbf{Q}\mathbf{R}, \quad (\text{B.11})$$

where  $\mathbf{Q}$  is  $NM \times NM$  and  $\mathbf{R}$  is  $NM \times 1$ . The only nonzero element contained in  $\mathbf{R}$  is the first element. Therefore, choosing the blocking matrix as

$$\mathbf{B} = \mathbf{Q}(:, 2 : NM)^H \quad (\text{B.12})$$

results in  $\mathbf{B}\mathbf{s} = \mathbf{0}$  as desired. This can be shown in the exact same manner as that done for the singular value decomposition.

### *Appendix C. Performance Measures*

The main goal of the radar is to detect the target amid noise, interference, and jamming. Target detection is a binary decision, either a target is present or there is not. Therefore, one of the obvious metrics for algorithm comparison is the decision test. This section addresses several of the common test statistics used for the binary detection problem, beginning with the one developed by Brennan and Reed in their pivotal paper [2]. Then the discussion transitions to two test statistics offering the desirable constant false alarm rate (CFAR) characteristic.

#### *C.1 RMB Test*

Continuing exactly as in Brennan and Reed's earlier work [2], the radar is cast in the light of a binary hypothesis problem. Either there is a target in the data and the hypothesis is  $H_1$  or there is not and the hypothesis is  $H_0$ . The data models are expressed by

$$\begin{aligned} H_0 : \chi_l &= \mathbf{n} \\ H_1 : \chi_l &= \alpha \mathbf{s} + \mathbf{n}, \end{aligned} \tag{C.1}$$

where  $\mathbf{n}$  is the colored noise with associated covariance matrix estimate  $\hat{\mathbf{R}}_l$ ,  $\mathbf{s}$  is the normalized steering vector, and  $\alpha$  is a random complex coefficient whose phase  $\phi$  is uniformly distributed between 0 and  $2\pi$ . The subscripted  $l$  emphasizes the fact that the data vector under consideration corresponds to a particular range cell. This is the exact same method used by Reed to develop the adaptive processor weights that maximized the probability of detection and subsequently the SNR [2] (introduced in Section 4.2).

Under these hypotheses and given complex Gaussian noise, the probability density functions of the incoming data vector  $\chi_l$  conditioned on the random phase of the target steering vector are

$$\begin{aligned} f_{\chi_l|H_1,\phi} &= \frac{1}{\pi^{NM} \det(\hat{\mathbf{R}}_l)} e^{-(\chi_l - \alpha \mathbf{s})^H \hat{\mathbf{R}}_l^{-1} (\chi_l - \alpha \mathbf{s})} \\ f_{\chi_l|H_0,\phi} &= \frac{1}{\pi^{NM} \det(\hat{\mathbf{R}}_l)} e^{-\chi_l^H \hat{\mathbf{R}}_l^{-1} \chi_l}, \end{aligned} \quad (\text{C.2})$$

where  $N$  is the number of spatial degrees of freedom and  $M$  is the number of temporal degrees of freedom such that the steering vectors  $\mathbf{a}$  and  $\mathbf{b}$  are  $N \times 1$  and  $M \times 1$  respectively. This produces a space-time steering vector,  $\mathbf{v}$  or  $\mathbf{s}$ , of size  $NM \times 1$ . The likelihood ratio test then becomes,

$$\Lambda_1 = \frac{\mathcal{E} \{ f_{\chi_l|H_1,\phi} \}}{\mathcal{E} \{ f_{\chi_l|H_0,\phi} \}} \underset{H_0}{\overset{H_1}{\gtrless}} \eta_1, \quad (\text{C.3})$$

where  $\mathcal{E}$  is the expected value operator with respect to the density of the random phase  $p(\phi)$ , and  $\eta$  is the threshold. Using the densities given above, the likelihood ratio test is

$$\Lambda_1 = I_o \left( 2|\alpha| |\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \chi_l| \right) e^{-|\alpha|^2 \mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \mathbf{s}} \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \quad (\text{C.4})$$

where  $I_o(x)$  is the modified Bessel function of the first kind. This is a Bayes optimal test. Isolating the signal power expression on the left side results in the Reed, Mallett, and Brennan (RMB) test [2],

$$\Lambda_1 = |\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \chi_l|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \left[ \frac{I_o^{-1} \left( \eta e^{|\alpha|^2 \mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \mathbf{s}} \right)}{2|\alpha|} \right] = \eta_1. \quad (\text{C.5})$$

Since the new threshold is a function of the covariance matrix and incoming data, it cannot be determined beforehand to achieve a specified false alarm probability. Hence, the test is not CFAR and is not practical for use as a test statistic.



### C.2 GLRT Test

The Generalized Likelihood Ratio Test (GLRT) was developed by Kelly in a paper on adaptive detection [13]. The test is given by

$$\Lambda_2 = \frac{|\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \chi_l|^2}{(\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \mathbf{s}) \left[ 1 + \frac{1}{2J} (\chi_l^H \hat{\mathbf{R}}_l^{-1} \chi_l) \right]} \stackrel{H_1}{\underset{H_0}{\geq}} 2J\eta_2, \quad (\text{C.6})$$

where  $\hat{\mathbf{R}}_l$  represents the interference covariance matrix estimate,  $\chi_l$  the primary data vector,  $\mathbf{s}$  the steering vector, and  $2J$  is the number of secondary data vectors used to calculate the covariance matrix estimate. Kelly shows this test does in fact possess the generalized CFAR property via the normalization by the signal dependent quantity in the denominator.

### C.3 AMF CFAR Test

The final CFAR test under discussion is one introduced just recently (1991) by Chen. The adaptive matched filter CFAR (AMF CFAR) test [4] is

$$\Lambda_3 = \frac{|\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \chi_l|^2}{\mathbf{s}^H \hat{\mathbf{R}}_l^{-1} \mathbf{s}} \stackrel{H_1}{\underset{H_0}{\geq}} \eta_3. \quad (\text{C.7})$$

This test statistic has also been referred to as the  $N_{\text{MSMF}}$  test statistic in articles by Melvin. Because the test provides a constant threshold that is a function of the probability of false alarm, it has an embedded CFAR characteristic. This is the source of the name.

The relationship between the threshold and  $P_{\text{FA}}$  is given by [4]

$$P_{\text{FA}} = e^{-\eta_3}, \quad (\text{C.8})$$

where  $\eta_3$  is the threshold.

This test was developed by Chen and Reed as an improvement of the GLRT test.

## Bibliography

1. Applebaum, Sidney P. and Dean J. Chapman. "Adaptive Arrays with Main Beam Constraints," *IEEE Transactions on Antennas and Propagation*, AP-24, No. 5:650-662 (September 1976).
2. Brennan, L.E. and I.S. Reed. "Theory of Adaptive Radar," *IEEE Transactions on Aerospace and Electronic Systems*, AES-9, No. 2:237-252 (March 1973).
3. Bronson, Richard. *Schaum's Outline of Matrix Operations*. New York, NY: McGraw-Hill, 1989.
4. Chen, W.S. and I.S. Reed. "A New CFAR Detection Test for Radar," *Digital Signal Processing*, 4:198-214 (October 1991).
5. Dudgeon, Dan E. and Russell M. Mersereau. *Multidimensional Signal Processing*. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc., 1984.
6. Goldstein, J.S. and I.S. Reed. "Subspace Selection for Partially Adaptive Sensor Array Processing," *IEEE Transactions on Aerospace and Electronic Systems*, 33, No. 2:539-544 (April 1997).
7. Goldstein, J.S. and I.S. Reed. "Theory of Partially Adaptive Radar," *IEEE Transactions on Aerospace and Electronic Systems*, 33, No. 4:TBD (October 1997).
8. Goldstein, J.S. and I.S. Reed. "A Tutorial on Space-Time Adaptive Processing," *IEEE National Radar Conference* (May 1997).
9. Graham, Alexander. *Kronecker Products and Matrix Calculus with Applications*. Ellis Horwood Ltd., 1981.
10. Haimovich, Alexander M. and Yeheskel Bar-Ness. "An Eigenanalysis Interference Canceler," *IEEE Transactions on Signal Processing*, 39, No. 1:76-84 (January 1991).
11. Harris, Fredric J. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, 66, No. 1:51-83 (January 1978).
12. Jaffer, A.G., et al. *Adaptive Space-Time Processing Techniques for Airborne Radars*. Contract F30602-89-D-0028, Fullerton, CA 92634: Hughes Aircraft Company, July 1991 (RL-TR-91-162).
13. Kelly, E.J. "An Adaptive Detection Algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, No. 1:115-127 (March 1986).
14. Melvin, Capt William L. "Nonhomogeneity Detection for Adaptive Signal Processing," *Proceedings of the 1996 IASTED International Conference on Signal and Image Processing* (November 1996).
15. Reed, I.S., et al. "Rapid Convergence Rate in Adaptive Arrays," *IEEE Transactions on Aerospace and Electronic Systems*, AES-10, No. 6:853-863 (November 1974).
16. Roberts, Richard A. and Clifford T. Mullis. *Digital Signal Processing*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1987.
17. Skolnik, Merrill I. *Introduction to Radar Systems, Second Edition*. New York: McGraw-Hill, Inc., 1980.
18. Sloper, D., et al. *Multi-Channel Airborne Radar Measurement (MCARM), MCARM Flight Test*. Contract F30602-92-C-0161, Box 1693, Baltimore, MD 21203: Westinghouse Electronic Systems, April 1996 (RL-TR-96-49, Vol 1 (of four), ADB211949).

19. Strang, Gilbert. *Linear Algebra and its Applications, Third Edition*. San Diego: Harcourt Brace Jovanovich, Inc., 1988.
20. Therrien, Charles. W. *Discrete Random Signals and Statistical Signal Processing*. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc., 1992.
21. Ward, J. *Space-Time Adaptive Processing for Airborne Radar*. Contract F19628-95-C-0002, Lexington, Massachusetts: Lincoln Laboratory, Massachusetts Institute of Technology, December 1994 (AD-A293032).

### *Vita*

Todd Hale was born 23 September 1971 in Dayton Beach, FL. Upon receiving his diploma from Spruce Creek High School in Port Orange, FL, he attended Milwaukee School of Engineering, Milwaukee, WI on an Air Force Reserve Officer Training Corps scholarship. Todd graduated with honors and received the Bachelor of Science degree in Electrical Engineering May 1993. His first Air Force assignment began 21 September 1993 at Offutt AFB, NE. His efforts as a B-52 Test Engineer for the 513<sup>th</sup> Engineering and Test Squadron earned him the Air Force Commendation Medal. In May 1996 he reported to the Air Force Insititute of Technology to begin work on a Master's degree in Electrical Engineering.

Permanent address: 125 Creek Crossing Rd.  
Daytona Beach, FL 32124-6606

- adaptive processing, 3-13
- FTS, *see* Factored Time-Space (FTS)
  - GSC, *see* Generalized Sidelobe Canceler (GSC)
  - maximum SINR, 4-5
  - STAP, *see* STAP
- ambiguity function, 3-8
- AMF CFAR test, *see* threshold detection,
- AMF CFAR test
- antenna array, 2-6
- element position vector, 2-8, 2-12
  - element power pattern ( $g(\theta, \phi)$ ), 3-18
    - example of, 3-35
  - element voltage pattern ( $f(\theta, \phi)$ ), 3-18
    - example of, 3-32
  - interelement spacing, 2-9, 3-2
  - number of elements, 2-8
  - pattern, 2-12, 2-14, 3-46
- Applebaum, 4-5
- beamforming, 2-4, 2-7, 2-12, 3-2, 5-8
- adaptive, *see* adaptive processing
  - element weighting, 2-9
  - example, 2-15
  - use of windows, *see* windowing
  - weighted delay and sum, 2-9, 2-13
- Bessel function, C-2
- Index*
- Bessel function, zero-order, 4-2
- Blackman Harris window, 3-42
- in FTS example, 5-13
  - specifications, 5-8
- blocking matrix (**B**), 4-6
- calculation of, B-1
- Boltzman's constant ( $k$ ), 3-36
- Brennan's rule, 3-29
- example of, 3-38
- clutter
- covariance, *see* covariance matrix (**R**)
  - model, *see* STAP, model, clutter
  - patch, 3-22
  - ridge, 3-27
  - slope ( $\beta$ ), 3-28
- Clutter-to-Noise Ratio (CNR), 3-24, 3-25
- example of, 3-32
- Coherent Processing Interval (CPI), 3-4, 3-12
- Constant False Alarm Rate (CFAR), 5-13, C-1, C-3
- coordinate system
- Cartesian, 3-3
  - STAP model, 3-3
- covariance matrix (**R**)
- estimation, 5-2
  - sample support for, 6-1

- of all undesired components, 3-40
  - of clutter, 3-26
    - rank of, 3-29
  - of jammer, 3-18
    - rank of, 3-20
  - of noise, 3-14
- Cross Spectral Metric (CSM), 4-27
- Cross Spectral Metric (CSM) model, 4-27, *see*
  - Generalized Sidelobe Canceler (GSC)
- analysis of, 6-1
- datacube, *see* Coherent Processing Interval (CPI)
- DFT matrix (as used in Doppler filter), 3-51
- Doppler
  - ambiguous, concept of, 3-28
  - filter, 5-8
  - in FTS, *see* Factored Time-Space (FTS)
  - normalized ( $\bar{\omega}$ ), 3-6
    - of clutter, 3-23
  - shift due to target ( $f_t$ ), 3-6, 3-10
- Earth, effective radius of, 3-22
- Factored Time-Space (FTS), 3-50
  - Doppler filter associated with, 3-50
  - example using MCARM data, 5-10
    - MCARM example using
      - non-homogeneity detection, 5-15
    - output ( $z_m$ ), 3-52
    - weight vector ( $\mathbf{w}_m$ ), 3-52
    - windowing, 3-51
- Fourier Transform, 2-3, 3-15
  - redefined, 2-3
- frequency
  - spatial ( $\vartheta$ ), 2-2, 2-5, 2-12, 3-8, 3-11, 3-28
    - of clutter, 3-23
  - temporal ( $f$ ), 2-2, 2-5, 2-10
- Generalized Inner Product (GIP), 5-5
- Generalized Sidelobe Canceler (GSC), 4-4
  - blocking matrix ( $\mathbf{B}$ ), 4-6
    - calculation of, B-1
  - full dimension
    - block diagram, 4-9, 4-13
    - model, 4-5
    - output ( $y$ ), 4-12
    - weight vector ( $\mathbf{w}_b$ ), 4-10
    - weight vector ( $\mathbf{w}_p$ ), 4-12
    - weight vector ( $\mathbf{w}_{GSC}$ ), 4-8
  - operation theory, 4-10
  - reduced dimension, 4-4, 4-11
    - block diagram, 4-14
    - output ( $y_r$ ), 4-13

- projection matrix, 4-13
- weight vector ( $\mathbf{w}_z$ ), 4-15
- transformation matrix ( $\mathbf{T}$ ), 4-6
- GLRT test, *see* threshold detection, GLRT
  - test
- grating lobes, 2-17
- grazing angle ( $\psi_c$ ), 3-23
- Hadamard product ( $\odot$ ), 3-41
- Hamming window
  - specifications, 3-55
- Hanning window
  - specifications, 6-9
- interference covariance, *see* covariance
  - matrix ( $\mathbf{R}$ )
- Jammer-to-Noise Ratio (JNR), 3-17
- Kronecker delta ( $\delta$ ), 3-14
- Kronecker product ( $\otimes$ ), 3-11, A-1
- likelihood ratio test, C-2
- Maximum Likelihood Estimation (MLE), *see*
  - covariance matrix ( $\mathbf{R}$ ), estimation
  - sample support for, 6-1
- MCARM, 1-3, 5-7
- Monte Carlo simulation, 6-1, 6-11
- noise covariance, *see* covariance matrix ( $\mathbf{R}$ )
- noise model, *see* STAP, model, thermal noise
- noise power per element ( $\sigma^2$ ), 3-14
- non-homogeneities
  - definition of, 5-5
  - detection of, 5-4
- plane wave, 2-1
  - frequency spectrum of, 2-3
- principal components, 4-29
  - method of, 4-15
- probability of detection, 4-2
  - maximized, 4-2
- probability of false alarm, 4-1, 5-13, C-3
- propagation velocity ( $v_p$ ), 2-5, 2-8, 2-10, 3-7
- Pulse Repetition Interval (PRI), 3-4
- Q-function, 4-2
- QR decomposition, B-3
- Radar Cross Section (RCS)
  - of clutter patch, 3-25
- receiver losses ( $L_r$ ), 3-18
- Reed's rule, 5-3
  - in CSM algorithm, 6-1, 6-9, 6-11, 6-13, 6-17, 6-22
- RMB test, *see* threshold detection, RMB test
- Sample Matrix Inversion (SMI), 5-3, 6-1

- sampling
  - of clutter, 3-31
  - spatial, 2-6
  - temporal, 2-7, 3-9
- secondary data, 5-2, 5-13
  - analysis of required support, 6-9, 6-22
- Singular Value Decomposition (SVD), 6-19, B-1
- SINR, 3-41
  - for CSM algorithm
    - fully adaptive, 6-2
    - partially adaptive, 6-6
  - for FTS, 3-53
  - for tapered fully adaptive STAP, 3-42
  - maximized, 3-40
  - optimum (fully adaptive STAP), 3-41
  - using interference covariance matrix estimate ( $\hat{\mathbf{R}}$ ), 5-3
- SINR Loss ( $L_{\text{SINR}}$ ), 3-43, 3-53
- slowness vector ( $\alpha$ ), 2-2
- SNR
  - loss due to interference covariance matrix estimation, 5-3
  - obtaining maximum, 4-3, 4-5
  - optimum, 3-44
- Space-Time Adaptive Processing, *see* STAP
- space-time signal, 2-4, 2-13, 2-14
  - superposition of plane waves, 2-4
- space-time snapshot ( $\chi$ ), 3-12
  - in clutter model, 3-24
  - in jammer model, 3-17
  - in noise model, 3-14
- space-time steering vector, *see* steering vector
- spatial frequency, *see* frequency, spatial
- STAP, 3-1
  - block diagram, 3-44
  - CSM model, *see* Generalized Sidelobe Canceler (GSC)
  - FTS, *see* factored time-space (FTS)
  - fully adaptive, 3-38
    - weight vector ( $\mathbf{w}$ ), 3-40
  - GSC form, *see* Generalized Sidelobe Canceler (GSC)
  - model
    - clutter, 3-21
    - electromagnetic environment, 3-2, 3-4
    - jamming, 3-14
    - physical environment, 3-2
    - thermal noise, 3-13
  - steering vector
    - in jammer model, 3-16, 3-20
    - normalized space-time ( $\mathbf{s}$ ), 4-4



- space-time (**v**), 3-11
- spatial (**a**), 3-10
- temporal (**b**), 3-11
- temporal frequency, *see* frequency, temporal
- threshold detection
  - AMF CFAR test, C-3
  - GLRT test, C-3
  - RMB test, C-2
- wavelength ( $\lambda$ ), 2-2, 2-10
- wavenumber vector (**k**), 2-3
- wavenumber-frequency space, 2-3, 2-4
- weight vector (**w**)
  - describing entire CSM algorithm
    - fully adaptive, 6-3
    - partially adaptive, 6-7
  - tapered, 3-40
- whitening filter, 5-6
- windowing, 2-17
  - Blackman Harris, *see* Blackman Harris
  - window
    - Hamming, *see* Hamming window
    - Hanning, *see* Hanning window
  - in Doppler filter, 3-51

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1997	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Secondary Data Support and Non-Homogeneities in Space-Time Adaptive Processing		5. FUNDING NUMBERS		
6. AUTHOR(S) Capt Todd Benjamin Hale, B.S.E.E				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Electrical Engineering Department 2950 P Street Wright-Patterson AFB, OH 45433-7765		8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/GE/ENG/97D-06		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Capt William Melvin RL/OCSS 26 Electronic Parkway Rome, NY 13441-4514		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES None				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) <p>One of the primary problems with the application of Space-Time Adaptive Processing (STAP) techniques is secondary data support for the interference plus noise covariance matrix estimate. Reed has shown the required secondary data support to achieve performance within 3 dB of optimal SINR is approximately equal to twice the degrees of freedom (DOF) used in the algorithm. Reed proved this rule for Sample Matrix Inversion (SMI) techniques.</p> <p>A concern arises when applying this rule to a newer class of reduced dimension STAP algorithms that do not fall under the SMI umbrella. This thesis focuses on the Cross Spectral Metric (CSM) algorithm developed by Goldstein and Reed.</p> <p>Through Monte Carlo simulations, the thesis proves Reed's rule for sample support is not accurate in this case. Optimum SINR performance for the CSM algorithm was obtained by choosing the number of DOF in the algorithm equal to the dimension of the interference subspace. With this choice, the required sample support for the covariance matrix estimate is 2.5 times the DOF used in the algorithm. This relationship is only true when the number of DOF is equal to the interference subspace dimension.</p> <p>A second goal of the thesis determines the impact of non-homogeneities within the secondary data on the CSM algorithm. The Generalized Inner Product (GIP) detection scheme is then used to excise these non-homogeneities from the secondary data. The CSM algorithm was found to be susceptible to non-homogeneities. The use of the GIP successfully negated the impact on this algorithm.</p>				
14. SUBJECT TERMS space-time adaptive processing, adaptive processing, adaptive filtering, covariance matrix estimation, dimension reduction, clutter, jamming, homogeneous data			15. NUMBER OF PAGES 201	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	